Automatic Recognition of Multiparty Human Interactions using Dynamic Bayesian Networks

Alfred Dielmann
Centre for Speech Technology Research
University of Edinburgh
Edinburgh EH8 9AB, UK

Doctor of Philosophy
Centre for Speech Technology Research
School of Informatics
University of Edinburgh
2009

Abstract

Relating statistical machine learning approaches to the automatic analysis of multiparty communicative events, such as meetings, is an ambitious research area. We have investigated automatic meeting segmentation both in terms of "Meeting Actions" and "Dialogue Acts". Dialogue acts model the discourse structure at a fine grained level highlighting individual speaker intentions. Group meeting actions describe the same process at a coarse level, highlighting interactions between different meeting participants and showing overall group intentions.

A framework based on probabilistic graphical models such as dynamic Bayesian networks (DBNs) has been investigated for both tasks. Our first set of experiments is concerned with the segmentation and structuring of meetings (recorded using multiple cameras and microphones) into sequences of group meeting actions such as monologue, discussion and presentation. We outline four families of multimodal features based on speaker turns, lexical transcription, prosody, and visual motion that are extracted from the raw audio and video recordings. We relate these low-level multimodal features to complex group behaviours proposing a multistream-modelling framework based on dynamic Bayesian networks. Later experiments are concerned with the automatic recognition of Dialogue Acts (DAs) in multiparty conversational speech. We present a joint generative approach based on a switching DBN for DA recognition in which segmentation and classification of DAs are carried out in parallel. This approach models a set of features, related to lexical content and prosody, and incorporates a weighted interpolated factored language model. In conjunction with this joint generative model, we have also investigated the use of a discriminative approach, based on conditional random fields, to perform a reclassification of the segmented DAs.

The DBN based approach yielded significant improvements when applied both to the meeting action and the dialogue act recognition task. On both tasks, the DBN framework provided an effective factorisation of the state-space and a flexible infrastructure able to integrate a heterogeneous set of resources such as continuous and discrete multimodal features, and statistical language models. Although our experiments have been principally targeted on multiparty meetings; features, models, and methodologies developed in this thesis can be employed for a wide range of applications. Moreover both group meeting actions and DAs offer valuable in-

sights about the current conversational context providing valuable cues and features for several related research areas such as speaker addressing and focus of attention modelling, automatic speech recognition and understanding, topic and decision detection.

Acknowledgements

Foremost I would like to thank my supervisor Prof. Steve Renals for providing me the opportunity to work on my PhD at the Center for Speech Technology Research - University of Edinburgh. I would like to thank my cosupervisor Dr. Miles Osborne for the useful advice and all the members of my PhD committee Dr. Mirella Lapata, Prof. Mahesan Niranjan, and Prof. Gerhard Rigoll.

I wish to thank all the CSTR members for their advice and constructive comments during the preparation of this thesis. A special thank you to Dr. Mike Lincoln who helped me in numerous occasions proving to be a dear friend and sharing the office with me for almost five years. Many thanks to Avril Heron and Caroline Hastings who made all the administrative and practical issues during my stay at UoE very easy.

Thank you to all the members of the M4, AMI and AMIDA research projects who generously helped me during the preparation of this thesis and to all the members of the AMI-ASR team which kindly provided the automatic transcriptions used in this work. Thank you to the members of the ICCS/HCRC - University of Edinburgh, SPANDH group - University of Sheffield, ICSI - University of California, IDIAP - Ecole Polytechnique Fédérale de Lausanne, and MMK - Technische Universität München. Moreover I wish to thank Sabrina Pei-yun Hsueh not only for the long conversations about our research within the AMI/AMIDA project but also for her sincere friendship.

I particularly want to thank my lovely wife Giulia for her unconditioned support during this “adventure” and for being my eternal sunshine. Finally I would like to thank all my relatives, friends and their families for their faith and support: Cati, Ignazina, nonna Cenza, Alessia, Paolo, Luigi, Daniela, Alberto, Federica, Andrea, Elena, Caterina, Francesco, Daniela, Fulvia, Ivana, Valentina, Annachiara, Silviona, Alberto, Carlo, Federico, Marzia, Pietro, etc.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Alfred Dielmann*
Centre for Speech Technology Research
University of Edinburgh
Edinburgh EH8 9AB, UK)

Table of Contents

Chapter 1

Introduction

The use of statistical machine learning for the automatic analysis of multiparty communication is an ambitious and novel research area. Complex social events such as meetings are a notable example of unconstrained multiparty conversations and provide a fertile and challenging environment for the investigation of novel methodologies. Meetings form a major part of many professional activities, in which work is planned, problems are highlighted and solved, decisions are made, and knowledge is shared. Preserving and accessing the information in such meetings is an important task , to enable a deeper understanding of the content of meetings, to make links across meetings, to facilitate remote meeting participants, and to disseminate knowledge to people who did not attend a meeting. The multimodal information contained in a multiparty meeting could be recorded using multiple cameras and microphones, devices to capture handwritten notes and other varieties of recording equipment. However, simply recording a conversation does not correspond to understanding what went on, and any information access from meetings requires additional processing.

During meetings people interact at different levels, showing both individual intentions and complex group behaviours, across multiple communication channels such as voice, gestures, and drawings. Features corresponding to these communicative modalities, such as speech, gestures, handwriting and facial expressions, may be extracted from raw audio-visual recordings. These individual feature streams can then be integrated to enable the identification of important events, such as discussions, presentations, questions, and statements. A probabilistic framework may be

used to learn from annotated examples the mapping between low level feature patterns and abstract categories. In such a supervised approach, manually annotated data is used to train the statistical models and to assess their recognition performances on unseen data.

In this thesis we focus on the automatic analysis of multiparty human interactions, investigating automatic meeting segmentation and indexing both in terms of "meeting actions" and "dialogue acts". These are two different aspects of the same problem. Group meeting actions (such as discussions, monologues, and presentations) aim to represent the whole group intention, providing a coarse grained representation of the meeting structure. Dialogue acts capture individual communicative intentions, such as statements, questions, and suggestions, highlighting the fine grained structure of a dialogue. Meeting actions and dialogue acts are two related aspects of the same communicative process, thus their automatic recognition can be conceived as a single task performed at two different granularities. Dialogue acts model the discourse structure highlighting the relationships between single speaker intentions. Meeting phases model the same communicative process highlighting interactions between different meeting participants and showing overall group intentions.

These two tasks have an interdisciplinary nature, involving machine learning, quantitative natural language processing, signal processing and meeting modeling. We hypothesise that a similar methodology can be successfully applied to both tasks, employing similar evaluation schemes, sharing a common set of multimodal features, and adopting the same probabilistic framework. On this challenging research domain, we have investigated the use of probabilistic graphical models, in particular dynamic Bayesian networks. The ability to integrate knowledge of the problem into the model is one of the most attractive features of a graphical approach. Moreover the resulting systems follow a common elegant mathematical formalism which can be fully observed, can be easily adapted to similar tasks, and can be quickly developed and efficiently tuned.

> *We would like to discover if the integration of multiple data streams using Dynamic Bayesian Network models can be beneficial for the automatic recognition of multiparty human-human interactions.*

To address this question we have conducted several experiments on "automatic

group behaviour modelling", by developing a system to segment and structure multiparty meetings using a dictionary of five basic meeting action types: discussion, monologue, note taking, presentation, and presentation at the whiteboard. In order to discriminate between different meeting actions, a statistical approach based on DBNs was adopted to discover and model repetitive patterns of the communicative process. Since groups interact across multiple modalities, such as speech, prosody, and gestures, it is useful to extract a relevant set of multimodal feature streams related to prosodic content, lexical information, speech and visual related speaker activities. The resultant statistical modelling framework, a multistream DBN model, will then integrate the information carried by each feature stream, model individual feature dynamics in an unsupervised way, and learn group behaviours from a set of manually labelled examples. This framework was evaluated over a set of short multi-party meetings, and numerical experiments showed good recognition accuracies (about 90% correct recognitions).

In analogy with the automatic meeting action recogniser, we have developed a fully automated system to perform joint dialogue act (DA) segmentation and classification. Dialogue acts represent the function that utterances serve in a discourse, and can be regarded as the atomic units of the communicative process behind a conversation. Meeting actions analyse the same conversation but at a coarse level of resolution. Similarly to the meeting action recogniser this system is based on a set of continuous features, several specialised language models, and a switching DBN based infrastructure. DA segmentation and classification are carried out in parallel, and the graphical model is used to coordinate the entire recognition process. Our DBN based generative approach models a set of features, related to lexical content and prosody, and incorporates a trigram discourse language model and a weighted interpolated factored language model (FLM). The FLM, which is estimated from multiple conversational corpora, is used in conjunction with additional task specific language models. In conjunction with this joint generative model, we have also investigated the use of a discriminative approach, based on conditional random fields, to perform a reclassification of the segmented DAs. We have carried out experiments on two corpora of multimodal meeting recordings, using both manually transcribed speech, and the output of an automatic speech recogniser, and using different configurations of the generative model. Our results indicate that the system

performs well both on reference and fully automatic transcriptions.

The adoption of a DBN based infrastructure proved to be effective both on the meeting action and on the dialogue act recognition experiments, achieving good recognition accuracies, facilitating resource reuse, and optimising the development cycle. Similar features and modelling solutions were shared by both systems, simplifying the development process. Moreover the same methodology translates well to other domains.

Automatic meeting analysis and structuring is a partially unexplored research field, which has been addressed only in the last few years. The availability of an automated system to extract the meeting structure, both in terms of meeting actions and dialogue acts, will facilitate the browsing of raw meeting recordings and provide valuable information for several related tasks, such as automatic summarisation , topic detection and tracking, action item detection , decision detection , and participant influence detection. Group meeting actions are also useful to index meeting collections, control active sensors such as pan-tilt cameras, facilitate the automatic editing of meeting video footage , or to develop context aware and pervasive applications such as an intelligent meeting room which reacts to the communicative situations (for example, dimming lights during presentations, or automatically showing the whiteboard content magnified on the main projection screen). Moreover these approaches are not constrained to the meeting domain and can be adapted to other contexts. For example the automatic structuring of conversational speech in terms of dialogue acts can be beneficial for automatic speech recognition, machine translation and spoken dialogue systems.

The proposed features and the multistream DBN infrastructure could be easily exported to many other research domains such as audio-visual speech recognition, video-surveillance, automatic broadcast content classification, human activity detection, and multimodal computer interfaces.

1.1 Overview of the thesis

This thesis is structured as follows:

- **Chapter 2** provides a brief introduction to Bayesian Networks (section 2.2) and to probabilistic inference on graphical models (section 2.3). Dynamic

Bayesian Networks (section 2.4) extend the BN graphical formalism to complex time series or data sequences. Moreover DBNs provide a unified graphical notation and mathematical formalism useful to describe probabilistic approaches such as Hidden Markov Models (section 2.4.1), Factorial HMMs (section 2.4.2), Coupled HMMs (section 2.4.3), and Hierarchical HMMs (section 2.4.4). Two extensions of DBNs, multi-rate models and Switching DBNs (also known as Bayesian multinets), are presented in section 2.4.5 and 2.5 respectively. Probabilistic inference on DBNs using the *interface algorithm* is discussed in section 2.6, and two DBN related software toolkits are compared in section 2.7.

- **Chapter 3** outlines the meeting data collections adopted in our experiments: the M4 (section 3.2.1), ICSI (section 3.2.2), and AMI (section 3.2.3) meeting corpora. The M4 corpus has been adopted for the group meeting action recognition experiments reported in chapter 5, and the latter meeting corpora have been employed by the joint dialogue act recogniser of chapter 7. Both systems rely on a DBN based infrastructure and on two collections of multimodal features (sections 3.4.1 and 3.4.2). These collections include features from the four feature families discussed in section 3.3: prosodic (section 3.3.1), "speaker turn" (section 3.3.2), lexical (section 3.3.3), and visual features (section 3.3.4).

- **Chapter 4** introduces the group meeting action recognition task reviewing several related works on individual (section 4.2) and group action recognition (section 4.3).

- **Chapter 5** outlines a DBN multistream model for the automatic meeting action recognition. The proposed framework relates 3 multimodal feature streams (sections 3.4.1 and 5.4.1) to high level group meeting actions such as discussion, monologue, note taking, presentation and presentation at the whiteboard. Numerical experiments on the M4 corpus (section 3.2.1) comparing the DBN multistream approach (section 5.3) and a baseline system (section 5.2) are reported in section 5.4. Two variants of this DBN framework are discussed in section 5.4.5. Section 5.5 validates the multistream

DBN meeting action recogniser on three independent feature setups, and section 5.6 reviews the latest advancements in the field.

- **Chapter 6** introduces the joint dialogue act segmentation and classification task. A review of the main approaches and feature sets, which have been adopted for this task, can be found in section 6.2. Section 6.3 outlines several applications which can benefited from automatic dialogue act recognition.

- **Chapter 7** investigates a switching DBN infrastructure (section 7.6) for the joint dialogue act recognition task. The proposed framework (section 7.2) integrates: six word related continuous features (sections 3.4.2 and 7.3), a trigram discourse language model (section 7.4), and two factored language models (section 7.5). Experimental results both on the ICSI (section 3.2.2) and AMI (section 3.2.3) data are reported in section 7.7. Reclassification experiments using a Conditional Random Field DA tagger are discussed in section 7.8.

- **Chapter 8** presents the latest experiments on dialogue act recognition. Section 8.2 applies the switching DBN dialogue act recogniser to a novel 4 broad DA task (sections 8.2.1, 8.2.3, and 8.2.4), and section 8.3 investigates a discriminative approach to improve the training of factored language model.

- **Chapter 9** summarises the work done in this thesis.

1.2 Declaration of previous work

This thesis is based on the following previously published material:

- A. Dielmann and S. Renals. *"Recognition of Dialogue Acts in Multiparty Meetings using a Switching DBN"*. IEEE Transactions on Audio, Speech and Language Processing, vol. 16, number 7, pp. 1303-1314, September 2008 (*chapters 3, 6 and 7*)

- A. Dielmann and S. Renals. *"Automatic Meeting Segmentation using Dynamic Bayesian Networks"*. IEEE Transactions on Multimedia, vol. 9, number 1, pp. 25-36, January 2007 (*chapters 3, 4 and 5*)

- A. Dielmann and S. Renals. *"DBN based joint Dialogue Act Recognition of Multiparty Meetings"*. In Proc. IEEE International Conference on Acoustics Speech and Signal Processing, pp. 133-136, April 2007 (*chapter 7*)

- A. Dielmann and S. Renals. *"Multistream Recognition of Dialogue Acts in Meetings"*. In Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI-06). pp. 178-189, Springer, 2007 (*chapter 7*)

- M. Al-Hames, A. Dielmann, D. Gatica-Perez, S. Reiter, S. Renals, G. Rigoll, and D. Zhang. *"Multimodal Integration for Meeting Group Action Segmentation and Recognition"*. In Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI-05). Lecture Notes in Computer Science 3869, pp. 52-63, Springer, 2006 (*section 5.5*)

- A. Dielmann and S. Renals. *"Multistream Dynamic Bayesian Network for Meeting Segmentation"*. In Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI-04), Lecture Notes in Computer Science 3361, pp. 76-86, Springer, 2005 (*chapter 3*)

- A. Dielmann and S. Renals. *"Multi-Stream Segmentation of Meetings"*. In Proc. IEEE Workshop on Multimedia Signal Processing, pp. 167-170, October 2004 (*chapter 3*)

- A. Dielmann and S. Renals. *"Dynamic Bayesian Networks for Meeting Structuring"*. In Proc. IEEE International Conference on Acoustics Speech and Signal Processing, pp. 629-632, May 2004 (*chapter 3*)

Chapter 2

Dynamic Bayesian Networks

2.1 Introduction

In this thesis we are interested in automatically recognising complex human behaviours using multichannel audio-visual recordings. This challenging task can be subdivided in two simpler steps: feature extraction and statistical modelling. The raw recordings are initially preprocessed in order to extract a collection of multimodal features, providing valuable cues which could facilitate the recognition process and reducing the amount of data that needs to be processed. Then a probabilistic modelling framework takes care of associating the observed features to the high level symbols that we aim to recognise. A wide selection of probabilistic approaches can be adopted on that purpose, these include: Hidden Markov Models (HMMs) (Baum, 1972), Artificial Neural Networks (Bishop, 1995), and Support Vector Machines (Vapnik, 1995). In this thesis we are interested in investigating the adoption of graphical probabilistic models (Cowell et al., 1999; Jordan, 1998) such as Dynamic Bayesian Networks (DBNs) and switching DBNs (section 2.5). This chapter provides a brief introduction to graphical models in general, Bayesian Networks (BNs), and DBNs. Graphical models offer a generic graphical formalism and mathematical notation useful to describe a large number of statistical approaches (Bilmes, 2003), including: BNs, DBNs, HMMs, Kalman filters (Kalman, 1960), Principal Component Analysis (Pearson, 1901; Bilmes, 2003), and Conditional Random Fields (Lafferty et al., 2001). BNs aim at representing static random variables and thus static problems. DBNs extend this modelling framework to time-

series, allowing to model an arbitrary set of variables as it evolves over time. DBNs can be interpreted as a generalisation of basic HMMs, since they allow to factorise the hidden state-space in terms of a set of interconnected random variables (section 2.4). A similar factorisation can also be extended to the observations, enabling multistream feature processing (section 5.3).

Graphical models are a flexible and powerful methodology to implement complex probabilistic models. They are based on the union of graph and probability theory (Cowell et al., 1999; Jordan, 1998). The graph theory provides an intuitive unified view over different statistical approaches and a general purpose set of algorithms to "deal with the model", i.e. to perform probabilistic inference (sections 2.3 and 2.6), train the model parameter set, find the most likely sequence of hidden states, and generate random observations according to the model. Probability theory provides a consistent way to integrate the model's components and a convenient interface to the outside world (e.g.: instantiating probabilistic evidence from the input observations, integrating ad-hoc probabilistic language models, generating posterior probabilities). Probabilistic graphical models make use of nodes to represent random variables and arcs to encode conditional independence assumptions among nodes. This results in simplifying the model graph [1] and in integrating some a priori knowledge of the underlying problem into the model. Directed graphical models contain only arcs with a predefined orientation, undirected graphical models do not rely on specific arc orientations. Both Bayesian Networks and Dynamic Bayesian Networks are notable examples of directed graphical models. Undirected probabilistic graphical models are usually referred to as Markov Random Fields. Figure 2.1 shows an example of a Bayesian Network or Belief Network (BN), where:

- Directed arcs imply causality between random variables. For example an arc from node B to node A suggests that A is caused by B.

- Observable variables representing measured parameters are graphically depicted through unshaded nodes.

- Shaded nodes indicate latent hidden variables or hypotheses.

[1] A fully connected graph needs to be employed when independence assumptions between the variables cannot be made.

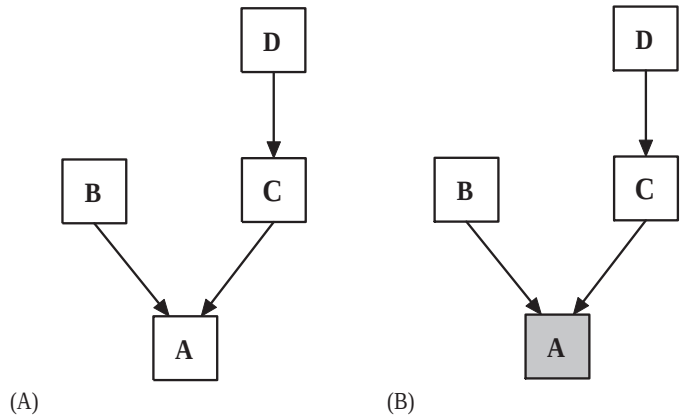


Figure 2.1: Two simple Bayesian Networks with the same graph topology: (A) is composed by 4 discrete hidden nodes, and (B) by 3 discrete hidden nodes and an observable discrete node.

- Continuous random variables are commonly represented with round nodes, being square nodes associated to discrete variables with a finite set of mutually exclusive states.

BNs are based on directed acyclic graphs (DAGs), thus all arcs should be carefully oriented avoiding cycles: paths with the same beginning and ending node cannot be constructed. Figure 2.3 shows some graphs which violate the DAG requirements needed by a BN.

Since nothing prevents instantiating the same BN multiple times, BNs can also be used to represent and model multiple sequential instances of a given set of random variables. However this will result in representing only local relationships among variables and disregarding their temporal dynamics. Dynamic Bayesian networks (DBNs) extend this concept adapting BNs to time-series or data sequences (section 2.4). Temporal evolutions of the observed data are explicitly modelled by inserting arcs between adjacent BN instances, thus forming chronological relationships between different temporal snapshots of the same nodes. Note that all these additional arcs need to be oriented from left (past) to right (present) according to the flow of time.

However having a model structure which changes according to its internal state is desirable in several applications. Complex problems can be formulated through a more compact and clear representation, reducing the number of free parameters and

the overall model footprint. Bayesian Multinets or switching DBNs (section 2.5) allow a dynamic change of the DBN graph topology, arcs can be enabled or disabled according to the state of some switching nodes (hypothesis nodes). This will result in enabling part of the graph only in presence of specific state configurations, and setting certain causal relationships among variables only when required.

The following mathematical notation will be used to describe BNs, DBNs and graphical models in general:

- Random variable and graph nodes will be indicated using capital letters such as: A, B, or X.
- Values taken on by these variables will be represented using lower case letters such as: $A = a$, $B = b$, or $X_t = x$.
- All the parents of a node C can be obtained through the function $Pa(C)$ and root nodes R have no parents $Pa(R) = \oslash$.
- Set of variable instances are indicated through temporal indexes such as $A_{1:t}$ or A_K if $K = 1:t$, assuming that all indexes start from 1. $X_{k:t} = x_{k:t}$ means that the node X from time k onward will assume values x_k, x_{k+1}, ..., x_t.
- Bayesian Networks (including both the DAG and the associated parameter set) are indicated with BN_k when part of a DBN, being k the frame index. BN_1 refers to the first temporal slice of the DBN and BN_t represents a generic BN adopted at time t.

2.2 Bayesian networks

A simple Bayesian network composed by 4 hidden nodes (A, B, C and D) and 3 oriented arcs (from B to A, C to A, and D to C) is shown in figure 2.1(A). The arc $\vec{BA}$ encodes a dependence relationship between B and A suggesting a causal relation from B to A. A similar discourse applies to $\vec{CA}$ and $\vec{DC}$. The DAG of a BN encodes several conditional independence relationships, for example nodes are independent of their ancestors given their parents. This results in being C independent of A given D:

$$P(C = c \mid D = d, A = a) = P(C = c | D = d) \tag{2.1}$$

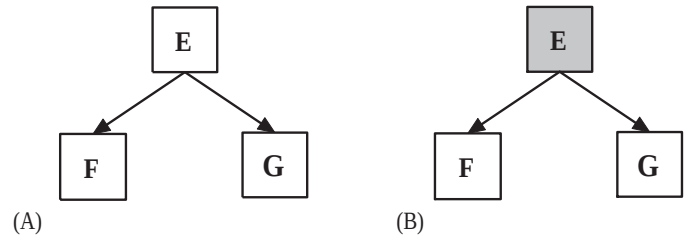


Figure 2.2: Other two simple Bayesian Networks with the same graph and 3 discrete nodes: node E is hidden (A), E is observable (B).

and being A independent of D given C:

$$P(A = a \mid C = c, D = d) = P(A = a | C = c). \tag{2.2}$$

Moreover since C is a hidden node with an ingoing arc $\vec{DC}$ and an outgoing arc $\vec{CA}$, A and D are conditionally dependent given C. Since node A is hidden, the lack of direct connections between B and C implies that they are conditionally independent given A.

Figure 2.1(B) shows a BN with the same graph topology of 2.1(A) but defining the node A observable rather than hidden. All the causal relationships (B and C cause A, C is caused by D) are still valid, but B and C are no longer conditionally independent.

The observable variable A has two competing causes B and C, and either of them is sufficient alone to explain A; thus B and C are conditionally dependent through A. For example, if A represents the evidence of a car breakdown and B, C are two independent explanations for A, e.g.: an "empty fuel tank" and a "broken battery", B or C are unrelated causes and either of them is sufficient to explain the car breakdown. Given the entire population of car breakdowns and using the proposed model to explain them, it will be evident that an empty fuel tank makes a broken battery less likely and vice-versa, even if these two events are clearly unrelated. This behaviour, usually referred as the "selection bias" or the "explaining away" phenomenon (Wellman and Henrion, 1993), appears when two or more causes are sufficient to explain a single observation.

Consider the two BNs in figure 2.2: if E is hidden (figure 2.2(A)) F and G are conditionally dependent; whenever E is observable (figure 2.2(B)) the two children will be independent. Finally, if node C of figure 2.1(A) is observable instead of

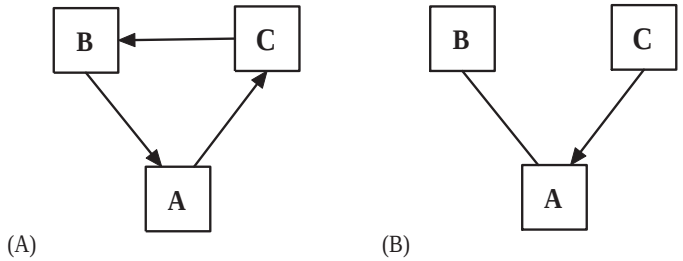


Figure 2.3: Examples of unacceptable Bayesian network graphs: (A) forms a cycle, and (B) contains an undirected arc ($A\overline{B}$).

being hidden, A and D will become conditionally independent given the observable node C.

Formally a BN β is a pair $\beta = (D,C)$ where: $D = (N,V)$ is the DAG containing a set N of nodes and V arcs, and C is a set of conditional probabilities distributions (CPDs). Each node in the graph is associated to a conditional probability distribution and C is a collection of all these CPDs:

$$C = \{p(N_i \mid Pa(N_i)) \mid N_i \in N\}$$

where $p(N_i \mid Pa(N_i)$ represents the probability of node N_i given all its parents $Pa(N_i)$ in the graph. The DAG D encodes the model topology and the parameter set C contains all the data-structures (associated to the graph D) needed to have a working implementation of the model. The model parameter set C is assumed to be time-invariant, and the same assumption can also be extended to DBNs and Bayesian Multinets (sections 2.4 and 2.5 respectively).

A unique joint probability distribution $P(U)$ for the whole BN β can be estimated through the product of all the conditional probability distributions defined by C:

$$P(U) = \prod_{N_i \in N} P(N_i \mid Pa(N_i)) \tag{2.3}$$

also including priors π_i from all the root nodes A_i contained by the graph D:

$$P(A_i|\oslash) = P(A_i) = \pi_i \quad \forall A_i \in N \ : \ Pa(A_i) = \oslash. \tag{2.4}$$

Note that a conditional probability distribution is not necessarily unique to a node, thus the same probability distribution can be shared by different variables with a

similar function or by subsequent snapshots of the same variable (section 2.4). Conditional probabilities of discrete nodes are often modelled through full or sparse multidimensional conditional probability tables (CPTs). Sometimes several relationships between nodes are know a priori, thus they can be excluded from the trainable parameter set and specified through deterministic rules or decision trees (section 2.7).

2.2.1 Gaussian Mixture Model

Continuous nodes are associated to continuous probability distributions such as Normal, Gamma, and Beta distributions, or even weighted combination of these distributions. Weighted mixtures of Gaussians are widely used statistical models, popular for their flexibility and mathematical tractability. The Expectation Maximisation (EM) algorithm (Dempster et al., 1977; Redner and Walker, 1984) can be used to estimate the parameters of these finite mixture models. This is an iterative procedure based on: calculating expectation values for the membership variables (assignments between data points and Gaussian components) of each data point, using the model parameters from the last maximisation step (E-step); and re-estimating the distribution parameters (mean, covariance, and class probability for each Gaussian) maximising the expected likelihood of the model on the entire dataset (M-step).

Gaussian Mixture Models (GMMs) represent a popular choice to implement the mapping between continuous observations and discrete states. Figure 2.4 depicts a common scenario: a n-dimensional vector of continuous features is associated to a continuous observable node Y, and a latent discrete variable X is the sole parent of Y. The mapping from continuous observations (Y) to discrete states (X) can be obtained through a GMM. Figure 2.4(B) shows the internal structure of a GMM by explicitly defining the "mixture of Gaussians" variable M: a discrete node with one state for each mixture component m. According to this graph, the conditional probabilities associated respectively to nodes Y and M are given by:

$$P(Y = y \mid X = j, M = m) = \mathcal{N}(y; \mu_{m,j}, \Sigma_{m,j}) \tag{2.5}$$

$$P(M = m \mid X = j) = C(m, j) \tag{2.6}$$

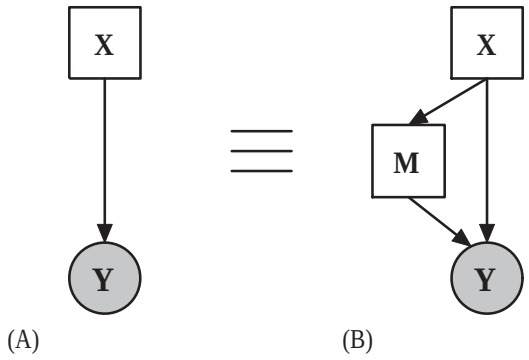


Figure 2.4: A continuous observable variable Y and its latent discrete explanation X are modelled through a Gaussian Mixture Model. The GMM can be implicitly (A) or explicitly (B) instantiated.

where $\mathcal{N}(y;\mu_{m,j},\Sigma_{m,j})$ represents a Gaussian density of mean $\mu_{m,j}$ and variance $\Sigma_{m,j}$ evaluated at y, and $C(m,j)$ the prior weight of each mixture component m conditioned by the current hidden state $X=j$. Assuming a n dimensional feature vector $y \in \Re^n$, each variance vector $\Sigma_{m,j}$ has exactly n components. The probability $P(Y=y \mid X=j)$ of observing the feature vector y given the current state $X=j$ can be estimated combining equations 2.5 and 2.6, and eliminating the hidden node M by summing over (section 2.3) all the mixture components $m=1,...,M$:

$$P(Y=y \mid X=j) = \sum_{m=1}^{M} C(m,j)\mathcal{N}(y;\mu_{m,j},\Sigma_{m,j}) \ . \tag{2.7}$$

Figure 2.4(A) shows the same GMM using an implicit notation in which M has been omitted. Both notations are acceptable even if the implicit one is the most common [2].

Note that node cardinalities (number of states for each discrete variable) and probability distribution formats (full table, decision tree, GMM, etc.) are not covered by the BN graphical formalism. These attributes, being part of the parameter set C, belong to the actual implementation of the model. Since both graphical infrastructure D and implementation details C are needed to define a working model infrastructure, all BN related toolkits (section 2.7) offer a formal language to fully specify both D and C.

[2] GMMs are usually considered a primitive Conditional Probability Density type, similarly to discrete conditional probability tables and decision trees.

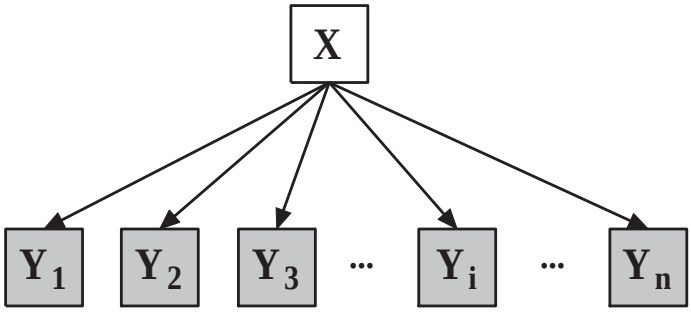


Figure 2.5: Graphical representation of the naïve Bayes model.

2.2.2 Naïve Bayes Model

The naïve Bayes model shown in figure 2.5 represents a simple but straightforward example of a BN model. This model was originally developed to facilitate medical diagnoses (Warner et al., 1961) inferring the most likely disease X from a set of observable features Y_1,Y_2,...,Y_n. An individual naïve Bayes network is associated to each disease X and the joint probability distributions $P(U)$ for each BN is estimated as:

$$P(U) = P(X, Y_1, Y_2, ..., Y_n) = \left(\prod_{i=1}^{n} P(Y_i \mid X) \right) P(X) \tag{2.8}$$

where $P(Y_i \mid X)$ represents the conditional distribution of each feature Y_i given the disease X. The posterior probability $P(Y_i \mid X)$ can be easily estimated from a training data-set containing the observed feature occurrences (symptoms) for each disease X. The naïve Bayes DAG implies that all the features Y_1,Y_2,...,Y_n are conditionally independent given X, but this assumption can be inappropriate.

2.2.3 Example

Richer and more complex models can be defined using the BN graphical formalism. Real daily-life problems can be easily encoded as a BN model, specifying well known relationships between the random variables, and thus incorporating precious human expertise about the problem. For example: an industrial bakery is willing to improve the quality of its bread, and the dough-rising process seems to be the key point to improve their bakery products. Years of practical experience suggested that: cooking time and temperature, percentage of water in the dough, and rising agent, are the most influential variables in the rising process. Plenty of successful and less

successful cases have been collected, thus a probabilistic model, able to forecast the outcome of a new variable combination (e.g.: bottom line of figure 2.6(A)), can be learned from the available examples. Moreover oven temperature and cooking time can be discretised, and human expertise can establish acceptable trade-offs between temperature and time (figure 2.6(A)), avoiding disastrous results such as overcooking. All this knowledge about the process can be integrated within a BN: figure 2.6(A) shows the causal relations between the variables involved; figure 2.6(B) formally depicts the DAG associated to the model; and figure 2.6(C) provides some further insights on CPTs and node cardinalities. Oven temperature E, cooking time T, and liquid content W are respectively mapped into B ($\mid B \mid = 4$: too low, low, medium and high), C ($\mid C \mid = 3$: short, medium and long) and D ($\mid D \mid = 10$) using three independent sets of GMMs. Node A ($\mid A \mid = 3$: acceptable, unacceptable trade-off, and unforeseeable results) integrates the deterministic relationships between B and C. Node R forecasts the outcome of the whole (rising) process given the state configuration of nodes P, D and A. A CPT of size $(2 * 10 * 4 * 3) * 2 = 480$ can be used to represent the likelihood of each of the 240 configurations of P,D,A given the two states of R.

During model parameter learning this CPT and the 4 GMM parameters (mean, variances and mixture weights) are learned from manually labelled examples. The outcome R of each cooking session is known, B and C have been manually determined, thus R, B and C are observable nodes during training. Node A, being deterministically estimated, lacks of any trainable parameter. The percentage of liquid content is discretised into 10 discrete states of the node D. The distribution of these 10 bins is unknown and their allocation is left to the training process, thus D is hidden even during training. It is likely that similar prediction accuracies can be achieved even with less than 10 classes, and thus a much smaller model footprint (e.g.: $\mid D \mid = 5$ will result in halving the whole parameter set), but this must be confirmed experimentally.

The amount of examples needed to train effective models depends on several factors like training data distributions and number of model's free parameters. Little control is available on the amount of training data and its distribution. However the model's parameter set can be carefully tuned, dimensioning each variable individually, and choosing the most appropriate state topology for every applica-

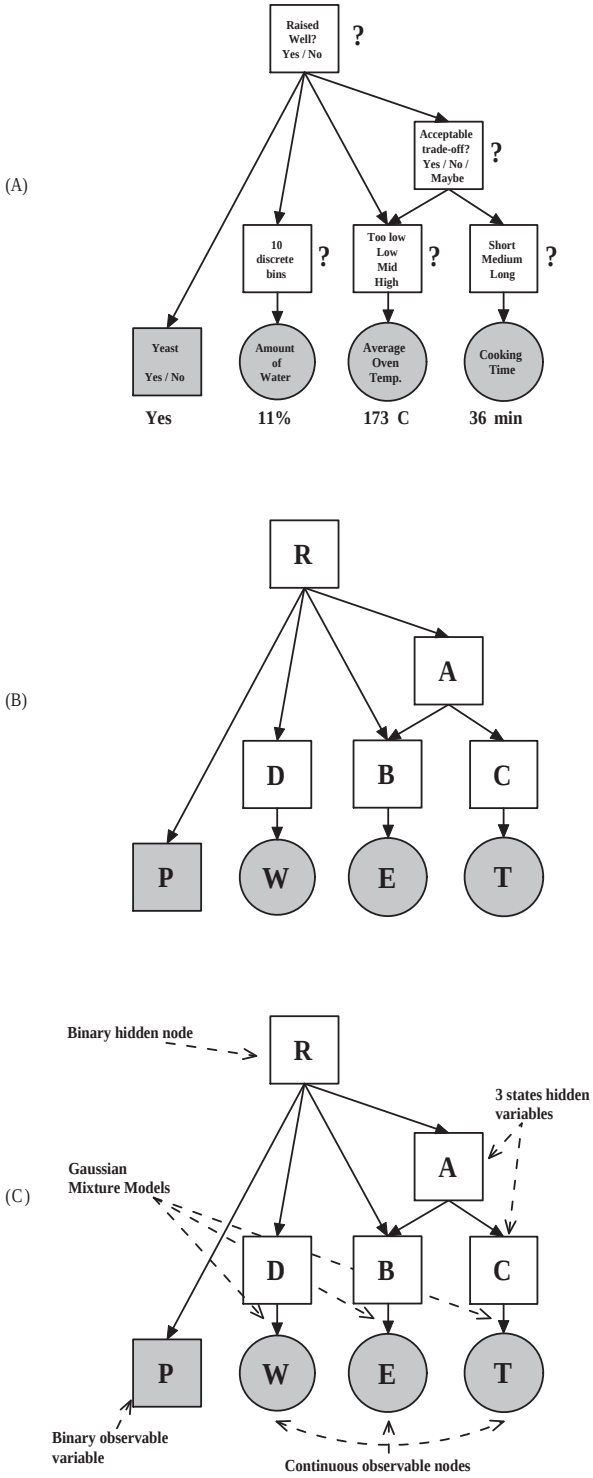


Figure 2.6: A practical problem and its Bayesian network representation: a simplified formulation of the problem (A), the resulting BN graph (B), and some further details about the model (C).

tion. The state space factorisation offered by BNs constitutes a powerful instrument to this end. Each variable can be dimensioned independently, and local dependences among variables can be individually investigated. Moreover conditional (in-)dependence relationships encoded into the graph provide a compact factorised representation for the joint probability of all the nodes involved. This representation may be exploited in order to reduce the computational effort required for probabilistic inference.

2.3 Probabilistic inference on Bayesian networks

Given a set of observable nodes, probabilistic inference allows to estimate the probability associated to the unobserved latent variables of the graph, suggesting the most probable explanation for a given observation. Probabilistic inference estimates the conditional probability $P(Q \mid Y = y)$ associated to a set of query variables Q given the evidence provided through the observable nodes Y. Probabilistic inference can be estimated from the joint distribution $P(U) = P(H,Q,Y)$ using Bayes theorem and summing out all the irrelevant variables H (marginalisation):

$$P(Q \mid Y = y) = \frac{P(Q,Y)}{P(Y)} = \frac{\sum_{H \neq Q,Y} P(H,Q,Y)}{\sum_{H \neq Y} P(H,Y)} \quad . \tag{2.9}$$

The query variable set Q can contain just a single node, or even all the hidden nodes in the graph. It is thus desirable to have an inference algorithm which maximises efficiency and resource reuse independently on the specific query.

Probabilistic inference also plays a key role on the model parameter learning task. Model's parameters are usually trained using expectation maximisation (EM) based algorithms, alternating likelihood expectation computations (E step) and maximisation of the estimated expectations (M step). Inference is the main tool to compute the likelihood expectation of each graph node. Since probabilistic inference is the key computation step both for parameter learning (model training) and model decoding, an efficient inference algorithm is highly desirable.

A naïve approach to probabilistic inference consists in answering to every possible query through "marginalisation": estimating the probability associated to the query alone by summing out all the irrelevant random variables. However given a fully connected model this procedure has exponential costs. Fortunately the DAG

of a Bayesian network provides a factorised representation of this task making marginalisation more efficient.

2.3.1 Variable elimination

Thanks to their structured relationships between variables, Bayesian networks offer an effective way to speed-up inference. The compact factored representation of the joint probability distribution may be exploited when efficient marginalisation is required. The key concept is to simplify this computation by reducing to the minimum the number of sums required, thus exploiting conditional independence between variables. This procedure (Zhang and Poole, 1994) exploits the principle of distributing sums over products during marginalisation by pushing sums as far as possible to the right end side of the equation.

For example, given the BN shown in figure 2.1, the probability that node A assumes the value a is given by:

$$P(A=a)=\sum_b\sum_c\sum_d P(A=a\mid B=b,C=c)\cdot P(B=b)\cdot P(C=c\mid D=d)\cdot P(D=d).$$

This product already includes the implications of equation 2.1. The sum over d appears only within the last two factors (D has only one outgoing arc $\vec{DC}$ and no parents), and the product may be rewritten as:

$$P(A=a)=\sum_b\sum_c P(A=a\mid B=b,C=c)\cdot P(B=b)\cdot\sum_d P(C=c\mid D=d)\cdot P(D=d). \tag{2.10}$$

This can be further simplified by imposing:

$$T_1(c,d)=\sum_d P(C=c\mid D=d)\cdot P(D=d)$$

and equation 2.10 will takes the form:

$$P(A=a)=\sum_b\sum_c P(A=a\mid B=b,C=c)\cdot P(B=b)\cdot T_1(c,d)$$

where T_1 was obtained eliminating the node D from the original graph. Dealing with more elaborate graphs, this variable elimination procedure may be applied iteratively until the graph is reduced to a single node and a unique term T_n. The computational cost associated to this algorithm is bounded by the number of variables

contained by the largest term encountered. Note that choosing the best sequence of "elimination variables" is a non trivial task and it is known to be a NP-hard problem.

Unfortunately variable elimination is query sensitive: the entire algorithm should be re-iterated for each set of query variables.

A query independent generalisation of the variable elimination procedure is provided by the Junction Tree (JT) algorithm. In order to discover an efficient factorisation, and therefore to provide an efficient inference algorithm, it is possible to proceed with some graph manipulations. The goal is to transform the directed graph into an undirected tree-shaped structure (junction tree) which efficiently supports the evaluation of multiple large queries. This tree shaped graph is built so that inference can be efficiently carried out using a message passing approach. The key point is to store in a tree shaped structure all the intermediate terms required during the inference computation and to reuse them as much as possible.

2.3.2 Moralisation

In the first instance, the graph needs to be converted into an undirected one through *moralisation*: unconnected parents are linked (married) together and arc orientations removed. For example the DAG associated to the BN in figure 2.7(A), after moralisation will be transformed into the undirected graph of figure 2.7(B). Arcs resulting from the moralisation (marrying of unconnected parent nodes) are depicted with heavy solid lines. These arcs are added in order to preserve the conditional independence assumptions made by the original BN. For example nodes A and B of figure 2.7(A) are conditionally dependent through their child C. Converting the DAG into an undirected graph, by simply dropping arc directions, would make A and B conditionally independent [3]. Note that the moralised graph introduces a less restricted factorisation than the original directed graph. Moreover marrying parents, and in general adding arcs to the graph, is a safe operation since it always leads to a "larger model" with fewer conditional independence assumptions.

[3] In any undirected graph, parents are always conditionally independent through their children.

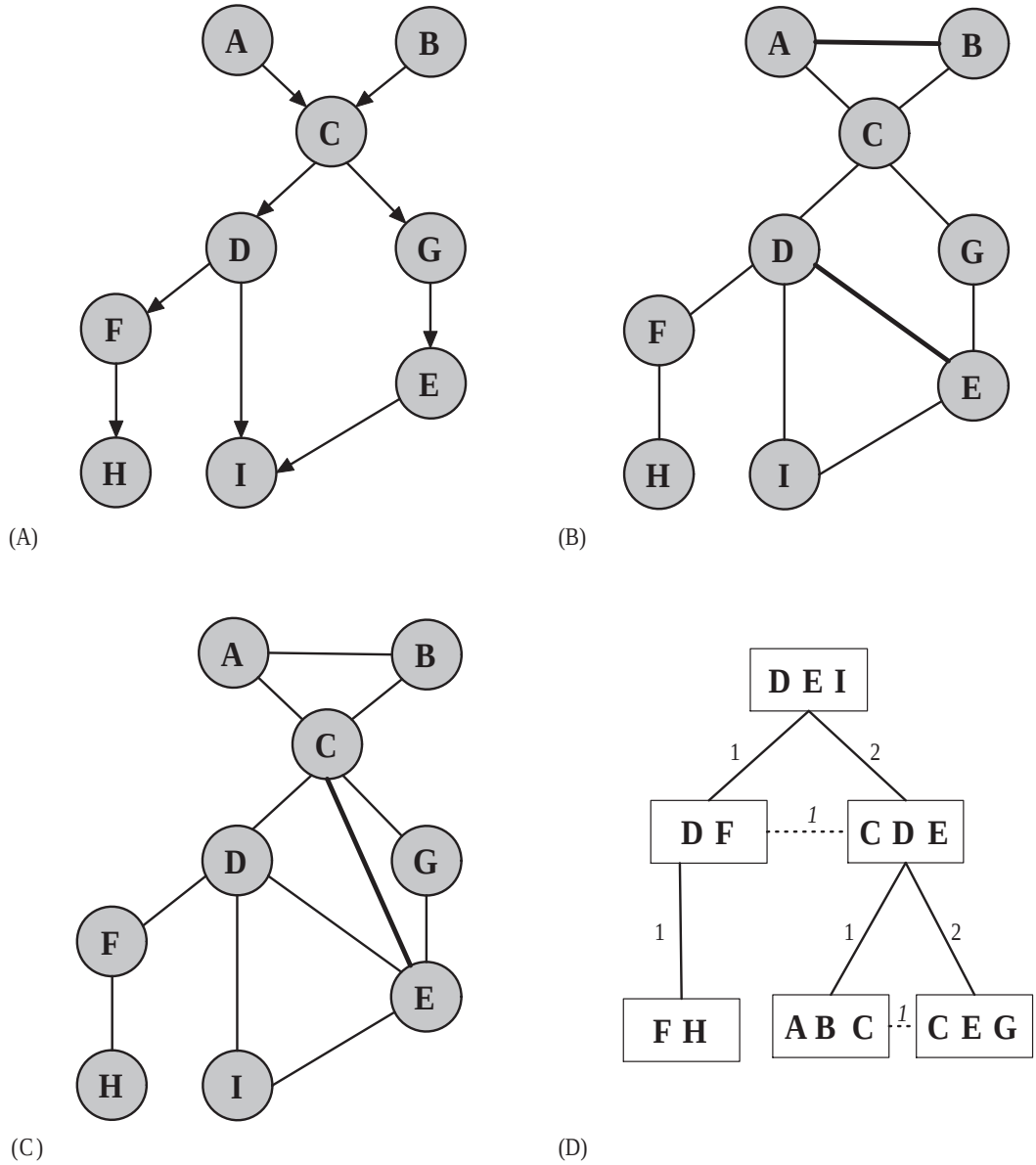


Figure 2.7: Construction of a Junction Tree: the original Bayesian network graph (A), moralised graph (B), triangulated graph (C), the resulting elimination graph (including dotted arcs) and Junction Tree (D).

2.3.3 Triangulation

The next step consists of performing the *triangulation* of the moralised graph: nodes are firstly ordered and then progressively removed from the graph after having all their neighbours (with higher ordering number) preventively connected. This procedure progressively eliminates all the graph nodes, replacing them with some additional "fill-in arcs". The triangulated graph is given by the original DAG augmented with all the new arcs introduced by the triangulation procedure. Note that all cycles from a triangulated graph with a length greater than three have a chord, thus a triangulated graph is often referred as a chordal graph. For example, given the undirected moralised graph in figure 2.7(B) and the node elimination order of figure 2.8(A), the resulting triangulated graph is shown both in figure 2.7(C) and 2.8(A). Node elimination orders are arbitrary: the same graph (figure 2.7(B)) can also be triangulated following the node sequence specified in figure 2.8(B) and thus obtaining a slightly different triangulation [4]. Since different orderings result in different final triangulations and the overall cost of probabilistic inference depends on the resulting graph, it is possible to choose orderings which are better than others.

A clique is a set of pairwise adjacent nodes forming a complete subgraph: every node is connected to every other node in the subgraph. For example nodes *A*, *B*, and *C* of the graph shown in figure 2.7(B) form a clique of size 3. A maximal clique is a clique not contained in any larger clique, for example nodes *A*, *B* form a clique but not a maximal clique as *ABC*. During the node elimination process a set of maximal cliques (also known as elimination cliques) is created. For example the triangulation of the graph in figure 2.7(B) using the elimination order of figure 2.8(A) will lead to the following sequence of elimination cliques: FH, DF, DEI, CDE, ABC, and finally CEG. While the nodes are clustered together into cliques, the resulting cliques could also be connected together in order to build a complete cluster graph. The intersections of variables in adjacent clusters of nodes are called "separator sets".

[4] Some triangulations cannot be obtained following an elimination order (Arnborg et al., 1987).

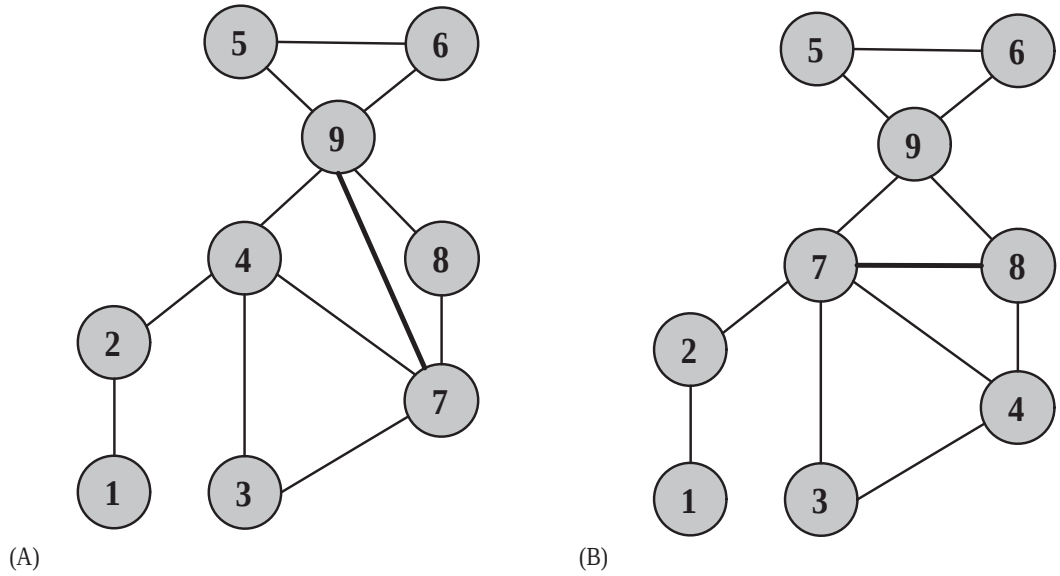


Figure 2.8: Example of how the choice of different elimination orders can result in different triangulated graphs. Nodes are progressively removed from the graph according to their index, from 1 to 9 in this example.

2.3.4 Junction Tree

The undirected graph built over the maximal elimination cliques can be converted into a tree of cliques. Arcs between cliques are firstly weighted according to the number of nodes in common among connected cliques (sizes of the "separator sets"). For example in figure 2.7(C), the separator set between cliques DEI and CDE is formed by two nodes: D and E; thus the arc between DEI and CDE has a weight of two. The resulting weighted graph is then converted into a tree by prioritising and leaving (as part of the tree graph) the connections with the highest weights. The maximum weight spanning tree obtained following this procedure is usually referred as a Junction Tree.

Figure 2.7(D) shows the junction tree associated to the elimination order of figure 2.8(A). Note that the original clique graph also contains two unitary weight arcs, between DF and CDE and between ABC and CEG respectively (dotted lines of figure 2.7(D)). These arcs were removed to form a tree so that each pair of cliques can be connected only through a unique path.

Once again different elimination orders often lead to different: cliques, separator sets, and thus clique trees. Since the cost of inference is exponential in the size of

the largest clique, it is desirable to build a junction tree with the smallest cliques, but this is known to be a NP-complete problem (Arnborg et al., 1987). Given a specific DAG, discovering a good triangulation for it can be a time-consuming operation. However this task should be performed only once. Then the resulting JT will be re-used over and over again during probabilistic inference.

Junction trees are built so that they satisfy the running intersection property: the intersection of any 2 cliques must belong to every clique on the path between them. For each pair of cliques c_1 and c_2 having a node X in common, each clique in the unique path between c_1 and c_2 should also contain the node X. Therefore, given a variable X, all its occurrences on the junction tree should appear as a connected sub-tree.

2.3.5 Message passing inference algorithm

Since in a Junction Tree all variable occurrences are grouped together, the information transfer is minimised when updating the model parameters. Moreover the JT offers a hierarchical view of the model: well delimited clusters of nodes (cliques from the triangulated graph) are organised into sub-trees according to the variables they have in common (separator sets).

As already observed in equation 2.3 the joint probability distribution $P(U)$ for the whole BN can be estimated as a product of conditional probability distributions. Having partitioned the graph into clusters of nodes $c \in \mathcal{C}$, the overall joint density $p(U)$ can be represented as a normalised product of potentials:

$$p(U) = \prod_{N_i \in N} p(N_i \mid Pa(N_i)) = \frac{1}{Z} \prod_{\forall c \in \mathcal{C}} \psi_c(X_c) \tag{2.11}$$

where each potential function $\psi_c(X_c)$ depends only on the variables X_c which are contained into the cluster c. Note that probabilistic evidence needs to be instantiated in all the clusters containing observable variables.

When node clusters match those suggested by one of the junction trees associated to the graph [5] the factorised density of equation 2.11 is also "decomposable". In such case a potential function ψ_c is associated to each clique $c \in \mathcal{L}$ of the junction

[5] A graph can have multiple junction trees and any of them can be used to this end. However, as pointed out in section 2.3.4, JTs with smaller cliques are more desirable than other.

tree, and the joint probability associated to the whole set of random variables U in the graph can be defined as the normalised product of potentials over the cliques c:

$$p(U) = \frac{1}{Z} \prod_{\forall c \in \mathcal{L}} \psi_c(X_c) \tag{2.12}$$

being Z given by:

$$Z = \sum_{X} \prod_{\forall c \in \mathcal{L}} \psi_c(X_c) \ . \tag{2.13}$$

Each potential function ψ_c depends only upon the nodes X_c contained by the clique c, and ψ_c is a non-negative continuous function or a multidimensional array in the discrete case. The factorised density of equation 2.11 can be converted into the decomposable density of equation 2.12 following these simple steps:

- build a Junction Tree for the original BN and allocate a potential function ψ_c for each clique $c \in \mathcal{L}$
- initialise all potentials ψ_c to the unity
- update each potential ψ_c using the cluster potentials from equation 2.11.

During this process all cluster potentials from the original joint probability density $p(U)$ of equation 2.11 that cover the variables in c are multiplied into ψ_c. Therefore during the update procedure of ψ_c, each factor (from equation 2.11) is multiplied into the potential ψ_c corresponding to the clique c with the same variables of the given factor.

The final step of the inference process (belief propagation) consists in the iterative application of a message passing algorithm. The goal of this procedure is to diffuse the observed information across the model, iteratively updating the hidden distributions until they are mutually consistent. Initially each clique c from the JT only knows its potential ψ_c and all its neighbour potentials. A message is then sent from c to all its neighbours, and each neighbour reacts combining its own local potential with the message (potential function) received from c. The combination between local potential and received messages allows each clique to re-estimate the marginal densities of its variables. Note that belief propagation is only guaranteed to be correct for trees, hence the need to convert a general graph into a Junction Tree.

This is frequently done using the "Hugin algorithm" also known as Jensen-Lauritzen-Olesen algorithm (Jensen et al., 1990), an approach based on the Shafer-Shenoy algorithm (Shafer and Shenoy, 1988) [6]. In the Hugin approach, a collection of non negative potential functions ϕ is associated to the Junction Tree, with ϕ_c being referred to as the *charge* associated with the clique c, and ϕ_s as the potential associated with the separator set s. Given two adjacent cliques c_1 and c_2 from the same JT, and if s is the separator set between c_1 and c_2, the initial charge is set as:

$$\phi_s = 1 \tag{2.14}$$

$$\phi_{c_2} = \psi_{c_2} \, . \tag{2.15}$$

Passing a message from c_1 to c_2 over the separator s will result in the following charge flow:

$$\phi_s^* = \sum_{c_1 \setminus s} \phi_{c_1} \tag{2.16}$$

$$\phi_{c_2}^* = \phi_{c_2} \frac{\phi_s^*}{\phi_s} \tag{2.17}$$

ϕ_s^* and $\phi_{c_2}^*$ will represent the new potentials on s and c_2 respectively, whenever ϕ_{c_1} and all the other potentials in the JT will remain unaltered. The factor:

$$\frac{\phi_s^*}{\phi_s} \tag{2.18}$$

represents the update ratio carried by the information flow from $c1$ to $c2$ along s. The information flow algorithm needs to be applied twice, so that two messages in opposite directions are passed along each arc (the separator set) of the JT. After all messages are passed, the joint probability density p_c of every clique c will be proportional to the corresponding potential ϕ_c.

In order to be consistent, this approach should obey a strict two phases propagation message passing protocol based on "message collection" and "message distribution". These two phases should comply with the following rule: a clique c is allowed to send (distribute) a message to a neighbour b only after having collected messages from all its neighbours (except b). Message distribution is subordinated to

[6] The Hugin algorithm is more time efficient than Shafer-Shenoy because running products required to compute clique beliefs are cached in memory rather than re-estimated over and over. However the Hugin approach involves the use of arithmetic divisions and the storage of both clique and separator potentials.

message collection, and a special clique needs to be selected from all the JT nodes, making this the starting point for the entire message propagation process. The JT root clique *r* is an intuitive choice and the two phase message passing algorithm can be applied as follows:

- the root *r* performs a collection
- each node *c* recursively invokes the collection on all its children (top to bottom)
- the leave nodes *b*, once reached by the collection request, will start passing up their potentials which are recursively propagated up to the root *r* (bottom to top)
- the root *r* starts a distribution process
- messages are recursively propagated from the root *r* down to the leaves *b* (top to bottom).

Therefore, given a generic node *c* and its child *b*, this process can be generalised defining two commands (one for each of the two phases):

- $Collect(c)$: recursively calls *collect* on each child *b* ($Collect(b)$) passing the resulting message up from *b* to *c*
- $Distribute(c)$: passes the message down from *c* to each child *b* recursively invoking *distribute* on each child *b* ($Distribute(b)$).

The whole message passing process can be started invoking $Collect(r)$ and completed with $Distribute(r)$. A simple but generic example is shown in figure 2.9: $Collect(r)$ results in messages being passed from cliques *b* to *c* and finally to *r*; $Distribute(r)$ results in passing messages in the reverse order, reaching the leaves at the end of the process.

Probabilistic inference on a static BN consists of two distinct phases: offline triangulation and JT online belief propagation. A maximum weight spanning tree (Junction Tree) can be obtained through a sequence of graphic manipulations: moralisation, triangulation and JT construction. This operation needs to be performed just once for a given DAG, then inference can be estimated online from the resulting JT

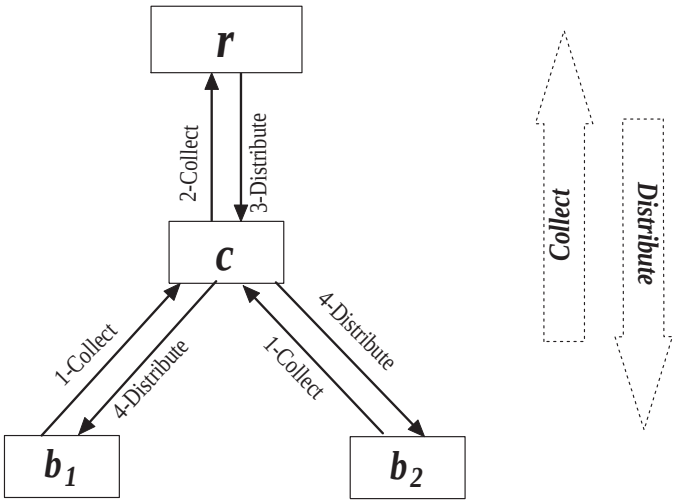


Figure 2.9: Sequence of *collections* and *distributions* during the message passing process.

by applying different observations and formulating multiple queries. At run-time beliefs can be propagated by following four simple steps:

- initialise clique potentials through equations 2.14 and 2.15
- instantiate probabilistic evidence through the clique potentials
- follow the collect-distribute message passing protocol, thus diffusing the observed information by applying the message passing algorithm of equations 2.16 and 2.17
- read the requested (through the initial query) conditional densities by normalising the obtained clique potentials.

Since exact belief propagation is computationally expensive (NP-hard in the largest clique), several approximate techniques have also been proposed, including loopy belief propagation, variational and sampling methods. An extensive review on the subject can be found in Murphy (2002a).

2.4 Dealing with dynamic Bayesian networks

Signal processing is inherently associated to evolving random variables, and most digital signal processing applications need to represent and model complex struc-

tures of time-dependent variables. Bayesian networks in their original formulation are limited to static random variables. Dynamic Bayesian Networks (DBNs) constitute an extension of static BNs which enables them to model complex time series or data sequences. In a DBN, a local BN is instantiated for each temporal slice [7], and the complete network is formed by adding interconnections between local BNs.

Each individual BN represents an instantaneous snapshot of the model taken at time t, describing the relations between different random variables within a single temporal frame. Complex temporal dynamics of the modelled variables can be represented by specifying additional arcs between nodes belonging to adjacent BNs, relating thus variables sampled at different instants of their temporal evolution. Hence a DBN is a set of static BNs interconnected by some additional causal links across slices, which explicitly represent the time flow. Although it is possible to define a specific BN for each temporal frame, a single generic BN is identically duplicated and applied to most of the frames, with the exception of the first and last temporal slices. Most DBNs can be represented using just three static BNs: a special BN_1 adopted only at time $t = 1$ known as the prologue, a generic template BN_t identically replicated $T - 2$ times, and the epilogue BN_T for the last frame T of the time-series. BN_1 deals with the model initialisation: estimating initial state configurations from priors, re-setting deterministic nodes, etc. BN_t acts as a generic template implementing the core model behaviours and determining how the state space will evolve in time according to the observed variables. Finally the epilogue BN_T, which is often equivalent to a generic template BN_t with the outgoing arcs removed, takes care of the last frame (temporal slice) of the data sequence. Although the vast majority of the DBN based models can be fully specified using at most three BN slices: prologue, template, and epilogue; some special approaches like multirate models form an exception (section 2.4.5).

Assuming a DBN which represents a process that is both stationary and Markovian, a further simplified "two slice temporal Bayes net" (2TBN) representation can also be adopted. In section 2.6 we will shows that this compact representation is convenient to perform probabilistic inference on DBNs. Processes modelled through DBNs are always stationary: conditional probability distributions are con-

[7] Assuming that all the variables are represented by discrete time samples taken at the same temporal instant and following an uniform temporal sampling and thus a fixed frame-length.

stant over time and thus all the model parameters are time invariant. Moreover the process is Markovian if the hidden variables X_t^i depend only on nodes from the current or the previous time slice $t-1$, and not on any older slice such as: $t-2$, $t-3$, etc.

All DBNs are based on the stationary assumption and most of them represent processes which are Markovian as well, thus allowing the adoption of a compact 2TBN representation. This can be constructed by unrolling two generic slices BN_t and BN_{t-1}, where each slice contains N nodes Z^i, $i = 1,...,N$, from the variable set Z. The resulting 2TBN graph forms an exhaustive representation for the whole DBN model. The conditional probability distribution $P(Z_t \mid Z_{t-1})$ of the variable set Z at time t given its previous configuration at time $t-1$ can be estimated as:

$$P(Z_t \mid Z_{t-1}) = \prod_{i=1}^{N} P(Z_t^i \mid Pa(Z_t^i)) \tag{2.19}$$

where $Pa(Z_t^i)$ are the parents of Z_t^i according to the DAG. Because of the initial Markovian assumption required by the 2TBN model, nodes $Pa(Z_t^i)$ belong only to the current slice BN_t or to the previous slice BN_{t-1} [8]. Equation 2.19 is also valid for the last frame T: in this case the 2TBN, obtained by joining a generic template $BN_t = BN_{T-1}$ with the epilogue BN_T, lacks of any outgoing arc. During the first time-frame $t = 1$ the two slice temporal Bayes net is reduced to the prologue BN_1. Since the initial slice BN_1 has no incoming arcs, node parents $Pa(Z_t^i)$ can only be found on the same slice BN_1, and all the hidden variables X_i are initialised according to their prior probabilities $P(X_i = j) = \pi_i(j)$.

The joint probability for the entire DBN can be obtained from equation 2.19 by unrolling the $\{BN_{t-1}\ ,\ BN_t\}$ model for T frames, and can be written as:

$$P(Z_{1:T}) = \prod_{t=1}^{T} \prod_{i=1}^{N} P(Z_t^i \mid Pa(Z_t^i)) \quad . \tag{2.20}$$

Similarly to static Bayesian networks (section 2.2) a Conditional Probability Distribution is associated to each individual hidden variable of the DBN. Note that parent-less hidden nodes from the prologue BN_1 are associated to prior state distributions; hard-coded deterministic relationships between variables, not being encoded as probabilistic CPDs, form an exception to a pure probabilistic framework.

[8] On multi-rate models (section 2.4.5) arcs are allowed to skip across slices, thus Z_t^i parents can be found even on more than two time slices.

Since DBNs represent stationary processes, the whole DBN parameter set (e.g.: probability distributions, Gaussian mixtures, mixture weights, and decision trees) is assumed to be time-invariant. Note that DBNs are "dynamic" because they extend the BN concept to time-series, but their parameters cannot change or evolve over time. When the model needs to change its behaviour according to the data temporal evolution, two possible strategies can be followed:

- introduce some additional variables to explicitly address the temporal evolution of the data, like the "counter structure" adopted in section 5.3.2;
- define one or more switching variables and convert the DBN into a "Bayesian Multinet" (section 2.5), as for the switching DBN dialogue act recogniser outlined in section 7.6.

A large number of probabilistic models can be re-formulated using the DBN graphical formalism such as Hidden Markov Models (Baum, 1972), Kalman filters (Kalman, 1960), and Input-Output HMMs (Bengio and Frasconi, 1995). The following subsections will initially outline the graphical formulation for a baseline HMM (section 2.4.1), and then move to more flexible approaches, such as Factorial HMMs (section 2.4.2), Coupled HMMs (section 2.4.3), and Hierarchical HMMs (section 2.4.4). Finally a brief overview of multi-rate models is provided in section 2.4.5.

2.4.1 Hidden Markov Models

Hidden Markov Models (HMMs) are one of the simplest examples of DBNs. Figure 2.10(A) depicts the formal DBN representation for a continuous observations HMM. Continuous observable feature vectors are modelled by nodes Y and hidden discrete states are represented by nodes X. The mapping between discrete states X to continuous observations Y is implemented using Gaussian mixture models (section 2.2.1). The same model is also shown in figure 2.10(B), where the DBN is "unrolled" T times and is ready to be applied to a sequence of T observations ($Y_{1:T} = y_{1:T}$).

This model can be still interpreted using the conventional Hidden Markov Model notation of Rabiner (1989) by defining: a state transition probability distribution

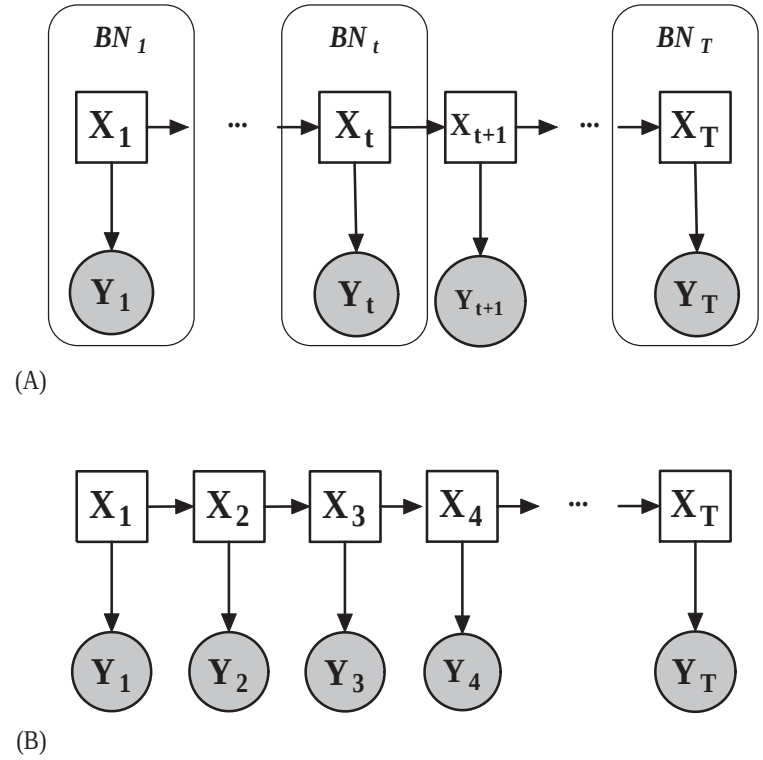


Figure 2.10: Dynamic Bayesian Network representation of a Hidden Markov Model applied to a time series of $\{1:T\}$ frames: a compact 3 slices representation BN_1, BN_t and BN_T (A); and the fully unrolled model (B).

A, an initial state distribution π, and a continuous observation probability distribution B. If the hidden state of node X at time $t-1$ is represented by $x_{t-1} = i$, the probability to change state at time t, reaching the new state $x_t = j$, is given by:

$$P(X_t = j \mid X_{t-1} = i) = A(i, j) \tag{2.21}$$

where A is the state transition probability matrix associated to the node X. Since A is a CPT, its rows should sum to unity and every element $A(i, j)$ is non-negative:

$$\sum_{j=1}^{|X|} A(i, j) = 1 \qquad i = 1, ..., |X| \tag{2.22}$$

$$A(i, j) \geq 0 \qquad i, j = 1, ..., |X| \quad . \tag{2.23}$$

In the prologue BN_1 the node X_1 lacks of any ancestor (incoming arcs), thus the probability of observing an initial state $x_1 = j$ is given by:

$$P(X_1 = j) = \pi(j) \tag{2.24}$$

where the prior distribution $\pi(j)$ should satisfy:

$$\sum_{j=1}^{|X|} \pi(j) = 1 \qquad j = 1, ..., |X| \quad . \tag{2.25}$$

The probability $P(Y_t = y \mid X_t = j)$ of observing the feature vector y given the current hidden state $X_t = j$ can be modelled as a weighted mixture of M Gaussians:

$$P(Y_t = y \mid X_t = j) = \sum_{m=1}^{M} C(m, j)\mathcal{N}(y; \mu_{m,j}, \Sigma_{m,j}) = B_j(y) \quad . \tag{2.26}$$

$C(m, j)$ represents the conditional prior weight of each mixture component m given the state j; $\mathcal{N}(y; \mu_{m,j}, \Sigma_{m,j})$ specifies a Gaussian density of mean $\mu_{m,j}$ and variance $\Sigma_{m,j}$ evaluated at y. Note that $P(Y_t = y \mid X_t = j)$ can also be indicated as $B_j(y)$ using the standard notation of the HMM literature (Rabiner, 1989; Young et al., 2006).

The GMM prior weights $C(m, j)$ should satisfy:

$$\sum_{m=1}^{M} C(m, j) = 1 \qquad j = 1, ..., |X| \tag{2.27}$$

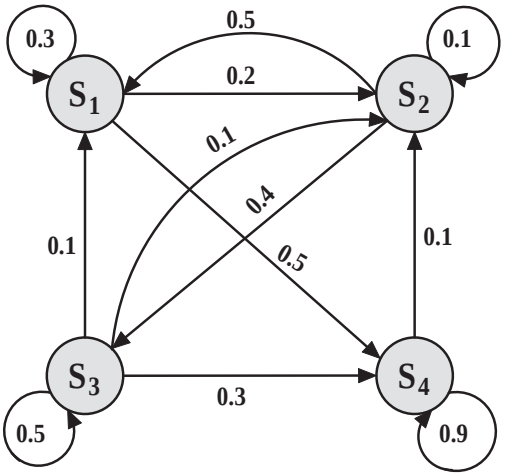
for all the $|X|$ hidden states of X. The joint distribution for a sequence of T temporal slices, considering the unrolled Hidden Markov Model of figure 2.10(B), can be obtained from equation 2.20:

$$P(X_{1:T}, Y_{1:T}) = P(X_1) \cdot P(Y_1 \mid X_1) \cdot \prod_{t=2}^{T} \{P(X_t \mid X_{t-1}) \cdot P(Y_t \mid X_t)\} \tag{2.28}$$

where the factors $P(X_1)$, $P(X_t \mid X_{t-1})$, and $P(Y_t \mid X_t)$ are respectively given by equation 2.24, 2.21, and 2.26. Therefore the joint probability over all possible state sequences $X_{1:T} = x_{1:T}$ can be written as:

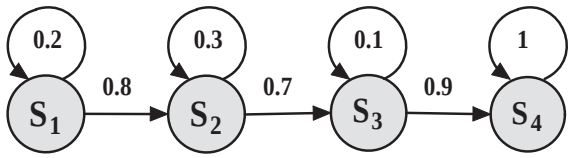
$$P(Y_{1:T} = y_{1:T}) = \sum_{x_{1:T}} \{\, \pi(x_1) \cdot B_{x_1}(y_1) \cdot \prod_{t=2}^{T} \{A(x_{t-1}, x_t) \cdot B_{x_t}(y_t)\} \,\} \quad . \tag{2.29}$$

The state transition probability matrix $A(i, j)$ encodes the HMM internal state topology, defining which state transitions are allowed and which state evolutions are forbidden. If all the elements of the state transition probability matrix $A(i, j)$ are greater than zero the HMM has a fully connected topology and each state can be reached from any other state. HMMs with a full topology are usually referred as "ergodic HMMs". Figure 2.11 shows a second example of an ergodic HMM.



$$A(i,j)=\begin{pmatrix} 0.3 & 0.2 & 0 & 0.5 \\ 0.5 & 0.1 & 0.4 & 0 \\ 0.1 & 0.1 & 0.5 & 0.3 \\ 0 & 0.1 & 0 & 0.9 \end{pmatrix}$$

Figure 2.11: State topology of an ergodic Hidden Markov Model and its associated transition probability matrix $A(i,j)$.



$$A(i,j)=\begin{pmatrix} 0.2 & 0.8 & 0 & 0 \\ 0 & 0.3 & 0.7 & 0 \\ 0 & 0 & 0.1 & 0.9 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 2.12: State topology of a left-to-right Hidden Markov Model, its associated upper diagonal transition probability matrix $A(i,j)$, and an initial state distribution of: $\pi(j) = [1\ 0\ 0\ 0]^{-T}$.

Not all state transitions are allowed by this topology (e.g.: S_1 to S_3), but since every state can be reached from any other state through a finite number of transitions (e.g.: S_1 to S_2 and finally to S_3), this is still an ergodic HMM. Another common HMM configuration is the "left-to-right state topology" (figure 2.12). According to this sequential topology, from each state i it is possible to remain in the same state i or to move to the next state $i+1$, but not to go back to any previously visited state j with $j < i$. The resulting transition matrix is upper tri-diagonal (figure 2.12) or upper diagonal when jumps of more than one state are also allowed (e.g.: $S1$ to $S3$). The initial state distribution $\pi(j)$ of a left-to-right HMM forces the model to start from S_1: $\pi(S_1) = 1$ and $\pi(S_i) = 0$, $\forall i \neq 1$. Adopting a DBN based implementation for a HMM, complex state topologies can be easily obtained during model training by: manually setting to zero some elements of the transition matrix $A(i,j)$ (forbidden

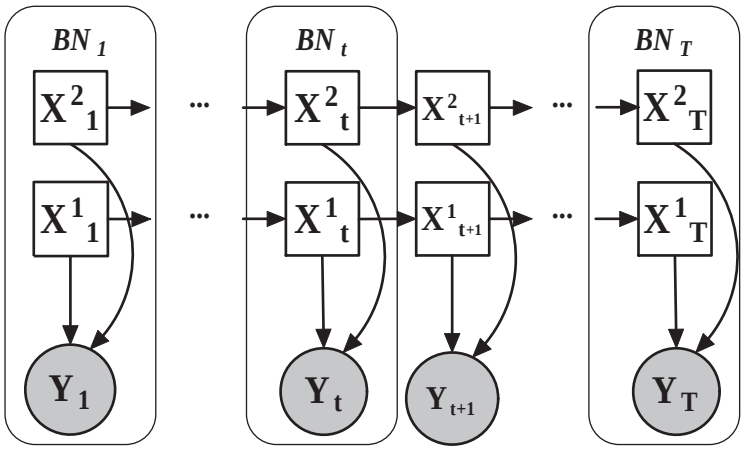


Figure 2.13: A factorial Hidden Markov Model with two chains (nodes X^1 and X^2).

state transitions) and then reestimating the model [9].

2.4.2 Factorial Hidden Markov Models

The classical formulation of a HMM consists in a single observation Y_t conditioned by a single hidden variable X_t. However the DBN representation allows to factorise the hidden state space into a set of hidden nodes X_t^1, X_t^2, ..., X_t^K (Jordan, 1998; Murphy, 2002a). Since each hidden node X_t^k represents a single individual aspect of the task, the resulting state space factorisation provides a clearer, well structured, and easily interpretable view of the underlying problem (Zweig and Russel, 1998; Zweig, 1998).

Sometimes different phenomena can be captured through the same sequence of observations, for example the speech signals not only convey verbal messages (through a set of utterances, words, and phonemes) but also some information about the speaker (such as gender and speaker size) and a very rich prosodic content. Therefore in some tasks a single observable feature vector needs to be shared between multiple HMMs. Factorial HMMs (Ghahramani and Jordan, 1997) are targeted to this class of problems. For example, Reyes-Gomez et al. (2003) adopted a Factorial HMM to separate multiple speakers using filter-and-sum microphone array processing (beamforming). Malkin et al. (2005) proposed an advanced FHMM approach to estimate the first two formants of a speech signal. Duh (2005) com-

[9] Any model parameter initialised to zero will remain so after reestimation.

pared several Factorial HMMs on a combined task of part-of-speech tagging and noun phrase chunking, showing that FHMMs allow to outperform the traditional method of tagging and chunking in succession.

An example of the Factorial HMM framework is provided by the DBN shown in figure 2.13, where two Markov chains composed by hidden states X_t^1 and X_t^2 share a single observable feature vector Y_t. Although X^1 and X^2 are a priori independent, they become coupled once we condition on the evidence Y. As outlined in section 2.2 this is due to the "explaining away" phenomenon. During the graph moralisation each node X_t^1 is married to X_t^2, making the two Markov chains conditionally dependent. Generalising to a factorial model with K chains, each chain X^k will be associated to its own transition probability matrix $A^k(i,j)$ and prior distribution $\pi^k(j)$ (defined as in equation 2.21 and 2.24):

$$P(X_t^k = j \mid X_{t-1}^k = i) = A^k(i,j) \tag{2.30}$$

$$P(X_1^k = j) = \pi^k(j) \ . \tag{2.31}$$

Therefore each chain X^k taken individually behaves like a conventional HMM. However, since all the hidden nodes X^k are conditioned on the same continuous observation Y, the conditional probability distribution associated to Y:

$$P(Y_t = y \mid X_t^1 = x_1, ..., X_t^k = x_k, ..., X_t^K = x_K) = B(y, x_1, ..., x_k, ..., x_K) \tag{2.32}$$

requires a large number of free parameters, one for each possible combination of its parents X^i:

$$\prod_{i=0}^{K} \mid X^i \mid \ . \tag{2.33}$$

Assuming for simplicity the same number of states N for each chain ($\mid X^i \mid = N$, $i = 1, ..., K$), the conditional probability density associated with Y contains N^K free parameters. Having K ergodic Markov chains with N discrete states each, the resulting probability transition matrices require a total of KN^2 free parameters. It is evident that a model with a large number of chains K can result in an extremely large state space, thus being intractable. However similar state configurations can be grouped together (state tying) to give a sparse representation for $P(Y_t \mid X_t^1, ..., X_t^K)$.

Note also that the whole factored model can be converted into a giant flat HMM with N^K hidden states (Cartesian product of all the K HMM chains) and a probability transition matrix with N^{2K} elements. However the resulting model is more

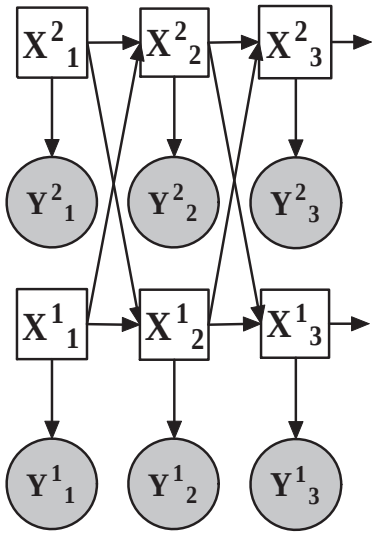


Figure 2.14: A coupled Hidden Markov Model with two Markov chains.

difficult to interpret and the overall computational costs are exponentially higher: a factorial HMM provides a useful factorisation of the state space if compared with a single flat HMM.

2.4.3 Coupled Hidden Markov Models

If multiple phenomena are observed on a single data stream (as outlined in section 2.4.2), the opposite situation can also be seen, in which a single process is jointly observable on multiple interdependent streams (Potamianos et al., 2004). For example speech consists not only of vocal sounds, but also of visually observable lip gestures (and an even larger number of concealed articulators such as tongue and velum). In particular Audio-Visual Speech Recognition (AVSR) aims at jointly modelling acoustic and visual observations. In this scenario visual information is used to enhance the acoustic content, leading to a more reliable and accurate speech recogniser (especially in presence of background noise).

Coupled HMMs (Brand et al., 1997) offer an intuitive modelling solution for this class of “multi-stream” related problems, and a detailed example about the application of coupled HMMs to the AVSR task is discussed by Nefian et al. (2002). Other applications of CHMMs include: speech driven realistic facial animation (Xie and Liu, 2007), human interaction modelling (Basu et al., 2001), and multi-channel

electroencephalogram data classification (Zhong and Ghosh, 2002).

Coupled HMMs similarly to factorial HMMs consist of multiple Markov chains. However each chain X^k is associated to its individual feature stream Y^k, and each hidden node X_t^k is coupled with the hidden states X_{t-1}^k of all the K chains [10]. Factorial HMMs (section 2.4.2) factorise the HMM state space over a set of hidden variables X_t^k, $k = 1,...,K$. Coupled HMMs push this concept further by factorising the feature space into K disjoint observation vectors Y_t^k. Each hidden node X_t^k is then responsible for its own feature subset Y_t^k, the dynamics of each Markov chain being influenced by the surrounding Markov chains X_t^l with $l \neq k$.

Figure 2.14 shows the graphical representation for a coupled Hidden Markov Model with two Markov chains, however the model can be easily generalised to K coupled chains. Prior state distributions $\pi^k(j)$ obey to the same definition given for factorial HMMs in equation 2.31. The conditional probability distribution associated to each continuous feature vector Y^k, given the discrete hidden state X^k, can be modelled through a GMM defined as in equation 2.26. Note that each Markov chain X^k and associated feature vector Y^k, are subject to their own independent set of Gaussian mixtures $P(Y_t^k = y \mid X_t^k = j) = B_j^k(y)$. Similarly to factorial HMMs each chain X^k is associated with its individual state transition probability matrix:

$$P(X_t^k = x_0 \mid X_{t-1}^1 = x_1,...,X_{t-1}^k = x_k,...,X_{t-1}^K = x_K) = A^k(x_0,x_1,...,x_k,...,x_K) \tag{2.34}$$

however these matrices span a much larger state space. Assuming N possible states for each of the K chains X^k, the resulting model will contain K transition matrices of size N^{K+1}. Therefore a fully connected coupled HMM can be intractable even for relatively small state spaces N and few concurrent streams K.

Coupled HMMs can be considered as a straightforward extension of plain HMMs for multi-stream problems, however their practical application is frequently restricted to small scale tasks with few coupled chains (Nefian et al., 2002) or highly constrained state spaces (e.g. $\mid N \mid = 2$) (Kwon and Murphy, 2000).

[10] Coupling can be relaxed in order to reduce the number of interconnections and thus the overall model complexity (Kristjansson et al., 2000; Basu et al., 2001).

2.4.4 Hierarchical Hidden Markov Models

Hierarchical Hidden Markov Models (HHMMs) (Fine et al., 1998) address problems where a complex state structure can be described using a hierarchical representation. For example, HHMMs have been employed to incorporate grammatical knowledge into Information Extraction models (Skounakis et al., 2003), to generate 2D illustrations from hand drawn sketches (Simhon and Dudek, 2004), to automatically structure long video sequences (L. Xie and Sun, 2003), and to learn musical structures from existing musical data sets (Weiland et al., 2005).

The state space is subdivided into a fixed number D of abstraction layers: the lowest layer which emits single observations (production state), $D-2$ intermediate layers emitting strings of observations, and the highest abstract layer which emits elaborate observation strings. A graphical representation for a 3 layer HHMM can be seen in figure 2.16. The lowest layer composed of the Markov chain X^1 is the only one directly related to the observable nodes Y [11]. The intermediate Markov chain X^2 staying on top of X^1 is used to supervise the lowest layer. The local state of X^2 influences the state transition probability of X^1 defining thus which states of X^1 are more likely and which transitions are not allowed (equation 2.35). The highest Markov chain X^3 plays a similar role on X^2 and then indirectly on X^1.

The state space for each layer includes a special "end-state": once this state is reached the control is returned to the higher Markov chain in the hierarchy. Therefore each layer X^d is invoked by its parent layer X^{d+1}, and X^d freely evolves within its own state space (following the state transition matrix selected according to the local configuration of X^{d+1}, X^{d+2}, ..., X^D) until the terminal state has been reached. The control is then given back to the parent layer X^{d+1} which updates its internal state and selects a new state transition matrix for X^d. Finally X^d is invoked with the newly selected state configuration and the whole procedure re-iterated until all the frames have been processed.

Nodes E^d take care of signalling to the upper layers when an end-state condition has been reached by X^d. The "end-state" node E^d also prevents the upper chains X^{d+1}, ..., X^D from changing their own internal state until the terminal state of X^d has been reached ($X^d = \text{end}$) and signalled ($E^d = 1$).

[11] This assumption can be relaxed by further generalising the HHMM and allowing middle layers to generate observations.

A HHMM can also be interpreted as a stack of Markov chains where the lowest layer, being the closest to the observations, takes care of modelling atomic changes in the data, and the middle and top layers define a structured set of abstract rules, capturing complex data behaviours. Control is passed recursively across layers preserving the calling context: higher layers call lower layers and the control is yielded back on reaching some special state configurations ("end-states").

HHMMs can be used to model complex generative processes. For example figure 2.15 shows the state transition diagram for the process required to generate observation strings according to the regular expression: $A \mid A(CDE)^{+}B^{+} \mid F^{+}(CDE)^{+}G$. The state structure is organised in 4 levels: 3 abstract levels and a production layer (nodes b). Each level has its private state space, and state transitions between adjacent levels are represented through dotted vertical connections. Each level of figure 2.15 corresponds to a Markov chain [12] of the HHMM (figure 2.16).

Processing starts from the root node $t1$ then follows the path to $t3$ or $t5$ until reaching a production state b. Since empty strings cannot be generated, horizontal state transitions (e.g. from $m1$ to $m2$) are not allowed before vertical transitions (from $m1$ to $b1$). Note that the shared state sequence $\{b5,\ldots,b8\}$, which is responsible for generating strings like $(CDE)^{+}$, can be reached both from $m2$ and $m6$.

In general HHMMs can be used to describe complex finite state automata with a fully probabilistic framework. Formally a HHMM is composed by 3 different types of layers: top, middle, and bottom layers.

Bottom layer Is constituted by the Markov chain X^1 and the end state nodes E^1. End nodes E are "turned on" whenever the evaluation of a given layer reaches a terminal state and the control needs to be returned to a higher layer. For example $E_t^1 = 1$ implies that the bottom Markov chain has reached an "end state" and the middle layer X^2 is going to take over X^1. The conditional probability associated to the node X_t^1 depends on the overall state configuration of all the higher layers $X_t^{2:D}$ and on the end node E_{t-1}^1:

$$P(X_t^1 = j \mid X_{t-1}^1 = i,\, X_t^{2:D} = k,\, E_{t-1}^1 = e) \;=\; \begin{cases} \widetilde{A}_k^1(i,j) & \text{if} \quad e = 0 \\ \pi_k^1(j) & \text{if} \quad e = 1 \end{cases} \tag{2.35}$$

[12] In practice the top two levels can be merged in a single Markov chain, by omitting the abstract states $t1$ and $t2$, and embedding their functions within the state-space of the layer immediately below.

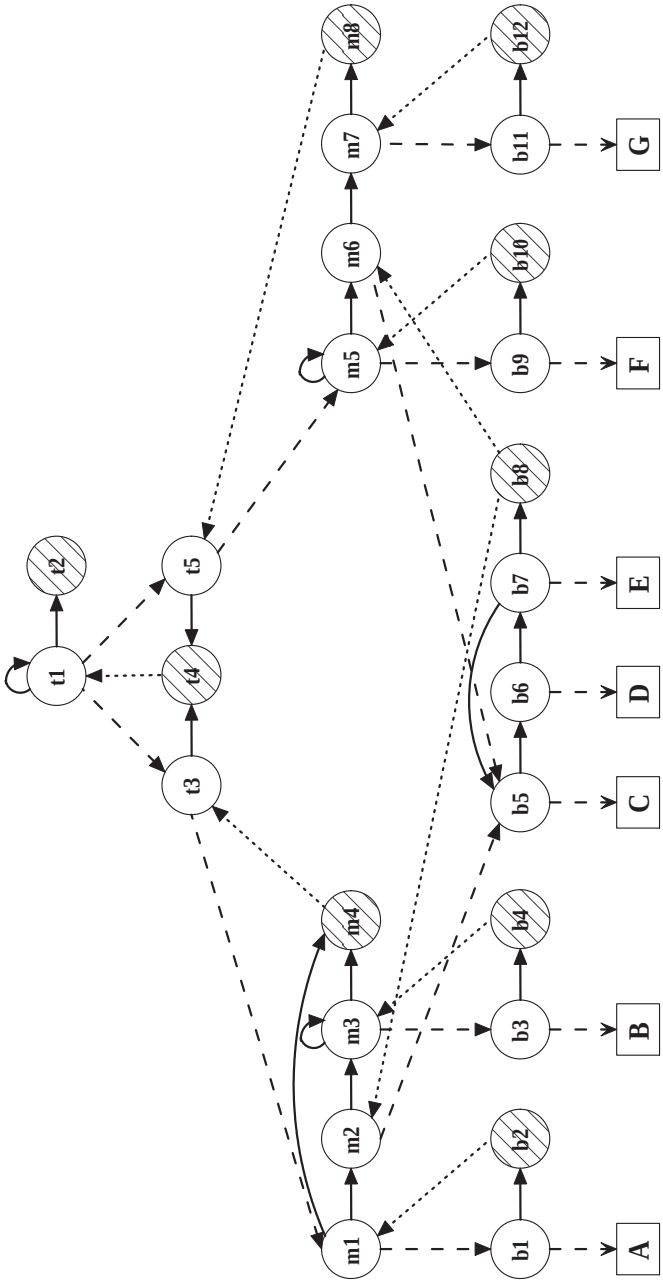


Figure 2.15: Hierarchical state transition diagram associated to the regular expression: $A \mid A(CDE)^{+}B^{+} \mid F^{+}(CDE)^{+}G$. States are disposed forming a hierarchy of 4 layers: nodes $t1$ and $t2$; nodes $t3$, $t4$ and $t5$; nodes m; and finally nodes b. Dotted arcs represent transitions to a lower layer and back to the calling node when an "end state" (shaded states) is reached. Observations (square symbols) can be generated only by the production states of the bottom layer (states $b1$, $b3$, $b5$, $b6$, $b7$, $b9$ and $b11$).

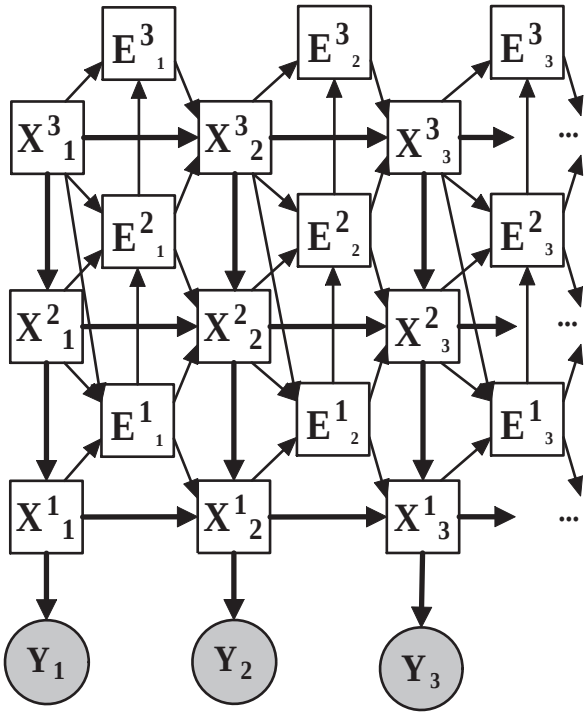


Figure 2.16: A Hierarchical Hidden Markov Model with three levels of hierarchy: the lowest level X_t^1, the intermediate Markov chain X_t^2 and the most abstract layer X_t^3. Each Markov chain X^d is associated to a set of "end-state" nodes E^d needed to coordinate the state-transition process across adjacent layers. In this example observations Y_t are directly related only to the lowest Markov chain X_t^1, although coupling between multiple layers through the observations can be easily implemented.

where $\widetilde{A}_K^1(i,j)$ represents the state transition matrix for the bottom layer given that the layers above X^1 are in state k, and $\pi_K^1(j)$ represents the prior state distribution adopted to re-initialise X^1 when an "end-state" $E_{t-1}^1 = 1$ has been reached. Note that equation 2.35 has been formulated under the hypothesis that $i, j \neq \text{end}$. If $A_k^1(i,\text{end})$ represents the conditional probability of reaching an "end state" from the overall HHMM configuration defined by $X_{t-1}^1 = i$ and $X_{t-1}^{2:D} = k$, the conditional probability associated to the node E^1 can be written as:

$$P(E_t^1 = 1 \mid X_t^1 = i,\, X_t^{2:D} = k) \;=\; A_k^1(i,\text{end}) \quad . \tag{2.36}$$

Each "end state", as formulated so far, once reached will not generate any continuous observation through Y. Unfortunately this is incompatible with

a DBN implementation since an observation needs to be generated during each frame. However the terminating property of "end states" can be incorporated as an attribute of conventional states. The new states can jointly generate an observation and turn on the "end-state" node $E_t^1 = 1$. The matrix $\widetilde{A}_k^1(i,j)$ refers to the DBN formulation with embedded terminal states, whenever $A_k^1(i,j)$ represents the same transition matrix explicitly including non-emitting terminal states. The probability of having $E_t^1 = 1$ is given by $A_k^1(i,\text{end})$, thus its complement $1 - A_k^1(i,\text{end})$ represents the probability of having $E_t^1 = 0$. The two equivalent state transition matrix formulations $\widetilde{A}$ and A should satisfy:

$$\widetilde{A}_k^1(i,j)(1 - A_k^1(i,\text{end})) = A_k^1(i,j) \tag{2.37}$$

because each state transition of A implicitly includes the terminal/non-terminal state probability of E_t^1, whenever this is modelled by $\widetilde{A}$ (and through nodes E_t) in the DBN based formulation.

Similarly to the baseline HMM, continuous feature vectors Y_t are mapped into discrete hidden states X_t^1 through a GMM (section 2.2.1) following the formulation outlined in equation 2.26. However the HHMM can be further extended adding conditional dependences between continuous features Y_t and middle-top Markov chains X_t^d. Similarly to Factorial HMMs (section 2.4.2) the intermediate layers will become conditionally dependent through their shared observations as in equation 2.32.

Middle layer The Markov chain X^2 together with the end nodes E^2 form the middle layer of the HHMM shown in figure 2.16. In general a hierarchical model with D layers has $D-2$ middle layer chains. For simplicity we refer to the generic middle layer chain using the index d. The main difference between bottom and middle layer chains consists in the presence of two additional arcs: between E_{t-1}^{d-1} and X_t^d; and between E_t^{d-1} and E_t^d. Therefore the probability of the hidden state X_t^d has a structure similar to equation 2.35, but is

also conditioned on $E_{t-1}^{d-1} = b$:

$$P(X_t^d = j \mid X_{t-1}^d = i, X_t^{d:D} = k, E_{t-1}^d = e, E_{t-1}^{d-1} = b) = \begin{cases} \delta(i,j) & \text{if} \quad b = 0 \\ \widetilde{A}_k^d(i,j) & \text{if} \quad b = 1,\ e = 0 \\ \pi_k^d(j) & \text{if} \quad b = 1,\ e = 1 \end{cases} \tag{2.38}$$

$b = 0$ implies that the lower layers have not reached an "end state" thus X_t^d is forced to remain in the same state until $b = 1$.

E_t^d can be turned on only when X_t^d is allowed to reach a terminal state, thus the conditional probability associated to E_t^d can be written as follows:

$$P(E_t^d = 1 \mid X_t^d = i, X_t^{d:D} = k, E_t^{d-1} = b) = \begin{cases} 0 & \text{if} \quad b = 0 \\ A_k^d(i, \text{end}) & \text{if} \quad b = 1 \end{cases} . \tag{2.39}$$

Top layer The most abstract layer (top of figure 2.16), similarly to the lower layers is composed by nodes X^D and E^D, however it does not depend on any further higher layer. Conditional probabilities of X^D and E^D follow a similar formulation to equation 2.38 and 2.39, however in absence of higher layers the conditioning term $X_t^{d:D} = k$ is no longer present:

$$P(X_t^D = j \mid X_{t-1}^D = i, E_{t-1}^D = e, E_{t-1}^{D-1} = b) = \begin{cases} \delta(i,j) & \text{if} \quad b = 0 \\ \widetilde{A}^D(i,j) & \text{if} \quad b = 1,\ e = 0 \\ \pi^D(j) & \text{if} \quad b = 1,\ e = 1 \end{cases} \tag{2.40}$$

$$P(E_t^D = 1 \mid X_t^D = i, E_t^{D-1} = b) = \begin{cases} 0 & \text{if} \quad b = 0 \\ A^D(i, \text{end}) & \text{if} \quad b = 1 \end{cases} . \tag{2.41}$$

During the first temporal slice (prologue) equation 2.35 and 2.38 should be rewritten as follows:

$$P(X_1^d = j \mid X_1^{d:D} = k) = \pi_k^d(j) \quad d = 1, \ldots, (D-1) \tag{2.42}$$

and the initial state distribution for the top layer Markov chain can be derived from equation 2.40:

$$P(X_1^D = j) = \pi^D(j) \quad . \tag{2.43}$$

All the conditional probabilities associated to the "end state" nodes E defined by equations 2.36, 2.39 and 2.41 apply unaltered to the first temporal slice $t = 0$. Note

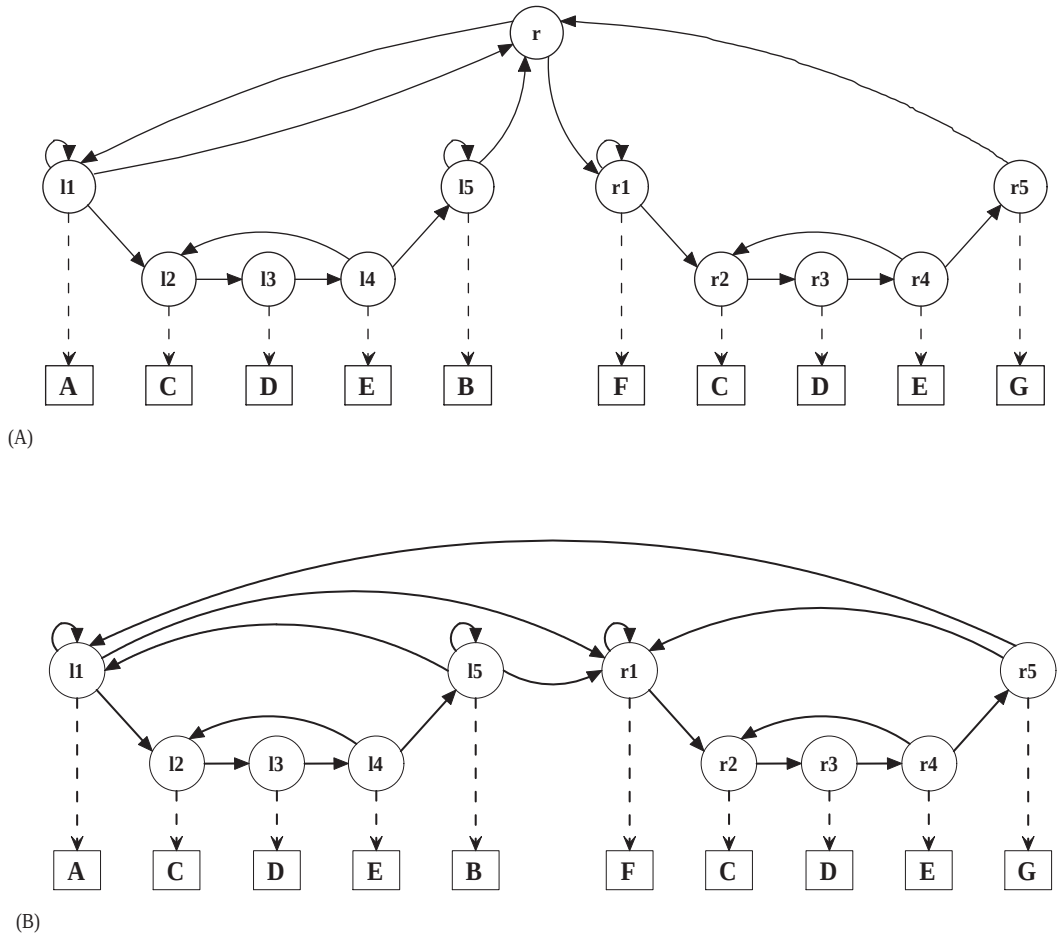


Figure 2.17: HMM state transition diagram equivalent to the Hierarchical HMM state diagram of figure 2.15 comprising: (A) a non generative root node r, and (B) in a DBN implementable formulation.

also that the last temporal slice (BN_T) is perfectly equivalent to a generic slice BN_t with the exception of having all the end state nodes E_T forcefully turned-on: $E_T^d = 1$ for all $d = 1, ..., D$.

Any HHMM can be converted into a flat HMM by visiting the original HHMM state transition diagram and creating an equivalent HMM state for every legal configuration of the hierarchical state $X^{1:D}$. For example given the HHMM state diagram of figure 2.15 the equivalent HMM state space is shown in figure 2.17(A). The abstract HHMM root node $t1$ of figure 2.15 is now the root node r of figure 2.17(A). Production states reachable through different paths are duplicated in the equivalent HMM state space. A highly structured HHMM state space, where long

state sequences are shared across different branches, can result in a much larger equivalent HMM with an higher number of free parameters to be learned.

For simplicity the equivalent state space of figure 2.17(A) contains a non generative node (the root r), which needs to be removed in order to have a DBN implementable HMM. This can be easily done replacing all the state transition from state x to y through r with a direct transition from x to y, as shown in figure 2.17(B).

The new equivalent HMM state space, if compared to the original HHMM of figure 2.15, clearly lacks a modular structure, having also a more difficult interpretation. Because of the duplicated state sequences, the training data-set should be larger in order to adequately cover both contexts. Alternatively equivalences between states can be imposed artificially by adopting a state-tying scheme.

Complex applications such as Automatic Speech Recognition can be intuitively formulated using a hierarchical structure, defining for example utterances, words with multiple phoneme pronunciations, phones with the surrounding phoneme context (in order to model coarticulation), and sub-phone state sequences. However this elegant and structured formulation has been rarely implemented as a genuine DBN HHMM based speech recogniser (Zweig and Russel, 1998; Bilmes, 2003). Instead, because of computational costs and memory requirements, the hierarchical state space is often converted into a large flat HMM with shared states. ASR oriented highly optimised tools, such as the Hidden Markov Model ToolKit (HTK) (Young et al., 2006), provide a computationally efficient framework to train (model parameter learning) and decode very large HMMs.

2.4.5 Multi-rate models

All the DBN models outlined so far have been fully specified using 2 or 3 BN templates (BN_1, BN_t, and eventually BN_T), but cases exist where a larger number of BN slices is required. For example “multi-rate models” (Çetin, 2004) allow the mixing of observations with different sampling periods, thus observation vectors are no longer available for every frame. Multi-rate processing represents a powerful extension of the DBN approach: features characterised by different sampling rates can be integrated into the model without the need of re-sampling, and processed preserving their natural temporal scale.

A simple multi-rate model is shown in figure 2.18. Let Y be the principal ob-

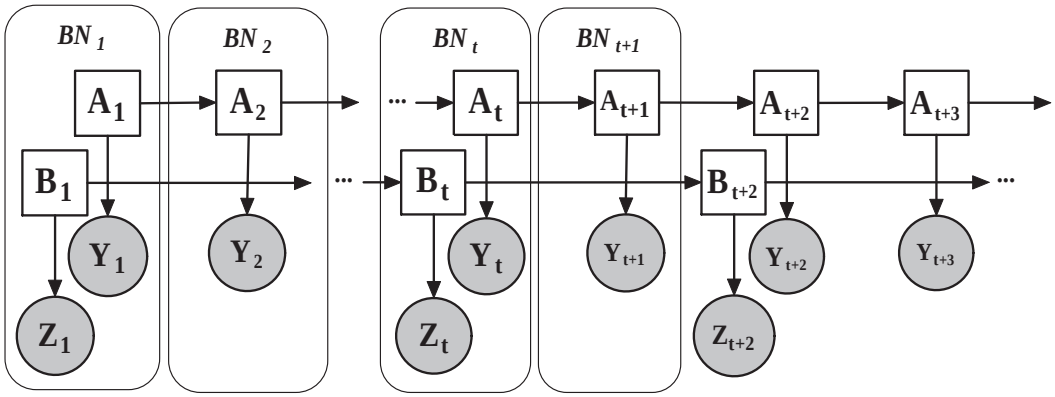


Figure 2.18: Multi-rate model composed by a 2 frame prologue (BN_1,BN_2) and a 2 frame template (BN_t,BN_{t+1}). The epilogue (BN_{T-1},BN_T) has been omitted.

servation sampled during each frame, and Z a feature vector with half the sampling rate of Y. Z and its associated hidden state B are present only every 2 frames, while Y and A appear in every time slice. As illustrated in figure 2.18 the prologue of this DBN is composed by 2 temporal frames and then 2 BNs, BN_1 and BN_2. Similarly the template and the epilogue are composed by 2 BNs each, respectively: BN_t, BN_{t+1}; and BN_{T-1}, BN_T. Therefore the resulting multirate DBN requires 6 different BNs in order to be fully specified.

2.5 Switching dynamic Bayesian networks: "Bayesian multinets"

Switching DBN models or Bayesian multinets (Geiger and Heckerman, 1996; Bilmes, 2000) are DBNs with multiple graphical topologies (DAGs) that can be selected according to the state of one or more switching random variables (hypothesis nodes).

For example the model depicted in figure 2.19 switches between two alternative topologies according to the state of the binary hypothesis node $H = [0,1]$. When $H = 0$ node C depends only on A through the arc $\vec{AC}$, as shown in figure 2.19(A). When the hidden nodes H switches to $H = 1$, as in figure 2.19(B), the arc $\vec{AC}$ is replaced by a new edge $\vec{BC}$, implying that node C now depends on node B. A compact formal representation for this switching model is given in figure 2.19(C): the arc $\vec{AC}$ is enabled only when $H = 0$ and the arc $\vec{BC}$ is active only when $H = 1$.

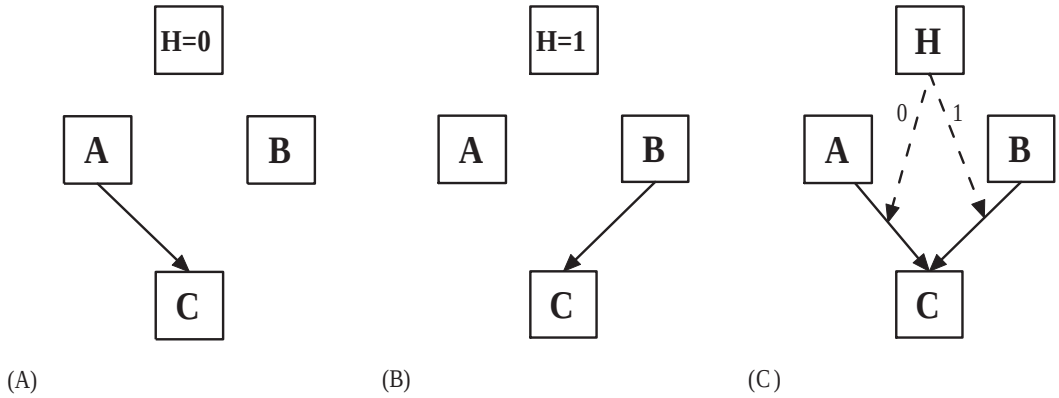


Figure 2.19: A simple Bayesian multinet: model's topology adopted when $H = 0$ (A), topology adopted when $H = 1$ (B), and a compact representation for both topologies (C).

The conditional probability distribution associated to the multinet in figure 2.19(C) is given by:

$$P(C \mid A,B) \;=\; P(C \mid A,H=0)P(H=0) \;+\; P(C \mid B,H=1)P(H=1) \qquad (2.44)$$

where $P(C \mid A,B)$ is also equivalent to the joint probability $P(A,B,C)$:

$$P(A,B,C) \;=\; \sum_{h=0}^{1} P(A,B,C \mid H)P(H=h) = P(C \mid A,B) \quad . \qquad (2.45)$$

The Bayesian multinet in figure 2.19(C) can be converted into an equivalent BN by instantiating both arcs $\vec{AC}$ and $\vec{BC}$ all the time and defining a unique CPT $P(C \mid A,B)$ instead of two CPTs, $P(C \mid A,H=0)$ and $P(C \mid B,H=1)$. However if all the three nodes have n states ($\mid A \mid = \mid B \mid = \mid C \mid = n$) the total number of free parameters is increased from n^2+n^2 (switching model) to n^3 (plain DBN).

Each local BN represents a distinct situation conditioned by a specific configuration of the hypothesis nodes. Although the model in figure 2.19(C) has only one switching variable and just two configurations (leading to two different local BNs), richer examples with more hypothesis nodes and more configurations can be easily constructed. Note that every random variable should appear on each local BN configuration [13]. For example the node A in figure 2.19(A) also appears within the BN of figure 2.19(B).

[13] Nodes can appear on demand on extended multinet models known as similarity networks (Geiger and Heckerman, 1996).

Although Bayesian multinets are usually associated to static models and "dynamic Bayesian multinets" (DBM) would represent their extension to time-series, the term DBM has already been employed to indicate Buried Markov Models (Bilmes, 2000). Therefore we refer to static multinets using the term "Bayesian multinets", proposing the new term "switching DBN" for dynamic implementations of multinet graphs.

Conventional Bayesian networks, if compared to a fully connected model, reduce the inference costs by enforcing conditional dependences between variables only when they are really needed. Bayesian multinets further extend this concept by allowing arcs to appear and disappear according to the state of some random variables. Specifying conditional dependences relationships, which are valid only on demand, simplifies the formulation of several problems (section 7.6) and further decomposes the state space, aiming at a smaller parameter set and at improving probabilistic inference. DBNs and switching DBNs cover the same model space: a multinet can be converted into an equivalent BN by adding arcs to the graph. However multinets are more intuitive to read than their DBN equivalents, and often result in savings, both in terms of computation time and memory requirements.

2.6 Probabilistic inference on a DBN

Exact probabilistic inference of discrete state DBNs is a task which can be addressed in several ways, often using the inference algorithm for static BN (section 2.3) as a sub-routine.

For example it is possible to transform the DBN into a HMM or a flat Bayesian network, and then adopt a well established approach to probabilistic inference such as the Junction Tree algorithm for BNs (section 2.3). As previously observed on factorial, coupled, and hierarchical HMMs, it is possible to convert a given DBN model into an equivalent HMM (Zweig, 1998) by building the Cartesian product over all the hidden discrete variables. A conventional forward-backward algorithm (Baum et al., 1970; Levy et al., 1996) can be then applied to the resulting HMM. However the product state space grows exponentially with the number of discrete nodes, leading to an intractable problem even for small DAGs. Alternatively the DBN can be converted into a large static BN. The dynamic model is unrolled for

the entire data sequence (T slices) building a unique large Bayesian Network (Murphy, 2002a). Then the static Junction Tree algorithm outlined in section 2.3 can be applied unaltered to this large BN. Unfortunately the resulting junction trees tend to have large cliques. For example a coupled HMM (section 2.4.3) with K Markov chains has at least K nodes for each clique (all the nodes with inter-slice connections). Therefore the resulting JT, built accounting for T frames, can become intractable even for small values of K. Moreover the whole inference algorithm, including the computationally expensive triangulation and JT construction process, needs to be reiterated every time that a new length T is required. However this simple approach, being already available at no additional cost, can be valuable for the validation of novel DBN specific inference algorithms.

Two DBN specific approaches for probabilistic inference are represented by the *frontier algorithm* (Zweig, 1996) and its evolution, the *interface algorithm* (Murphy, 2002a). These algorithms address the inference problem considering two adjacent temporal slices of the DBN, and may be regarded as a generalisation of the HMM forwards-backwards (FB) algorithm to DBNs. Both approaches focus on performing static JT BN inference on a generic 2TBN model (DBN equivalent representation introduced in section 2.4), starting from the first time-slice and then defining a procedure to go forward processing all the subsequent frames. In a HMM the FB algorithm works because conditioning on the hidden state X_t d-separates the past from the future (Murphy, 2002a). In a DBN the same kind of separation is given by the set containing all the hidden variables from the template BN_t: the frontier set Z_t. The *frontier algorithm* (Zweig, 1996) extends to DBNs the FB algorithm originally formulated for HMMs, by adopting a Markov blanket formed by the frontier set Z_t, and "sweeping" it forward and backward across the DBN. The Markov blanket (Pearl, 1988) of a given node X is the set of neighbours of X in the moral graph (Cowell et al., 1999). The blanket of X, including its parents, its children, and its children's parents [14], shields X from the rest of the BN. Therefore the Markov blanket of X contains all the information needed to explain any behaviour of X. At every step of the frontier algorithm, the Markov blanket Z_t d-separates all the nodes on the right from all the nodes on the left of the frontier set Z_t.

Murphy (2002a) *interface algorithm* further reduces the frontier distribution Z_t

[14] X children's parents are married to X during graph's moralisation (section 2.3.2).

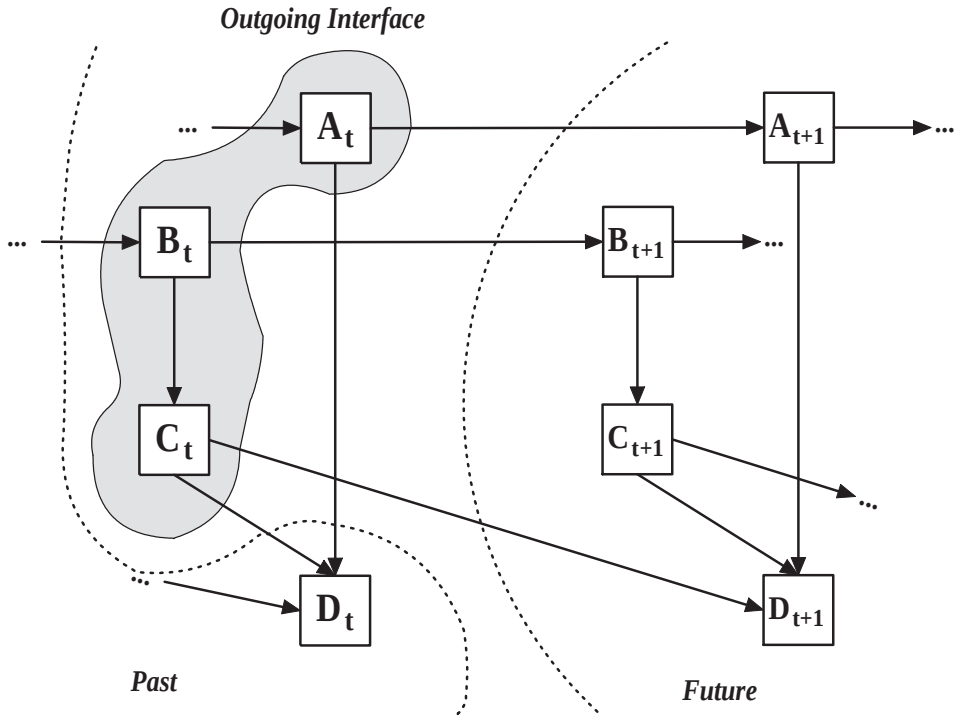


Figure 2.20: Two slice temporal Bayes net (2TBN) representation of a DBN composed by 4 hidden nodes. The outgoing interface I_t contains 3 nodes (A_t, B_t and C_t).

excluding all nodes that do not have any children in the next slice, and eventually exploiting further conditional independences encoded in the conditional probability distributions (but not explicitly shown in the graph). The outgoing interface I_t is a subset of the frontier Z_t containing all the nodes from BN_t with children in BN_{t+1}. Similarly to the frontier the outgoing interface d-separates the past from the future. The past consists of all nodes with a temporal index smaller than t together with the non interface nodes of frame t, and the future is represented by all the nodes belonging to the time-slices $t+k$ with $k \geq 1$. For example, given the DBN depicted as a 2TBN model in figure 2.20, the outgoing interface I_t is composed by the hidden variables A_t, B_t and C_t; the past is represented by node D_t and all the previous slices $t-1$, $t-2$, ..., 1; and the future includes time-slices $t+1$, $t+2$, ..., T. In analogy to the outgoing interface, the incoming interface can be defined as the set of nodes from BN_t with parents in BN_{t-1}. Usually the outgoing interface is not bigger than the incoming interface , for example the 2TBN in figure 2.20 has an

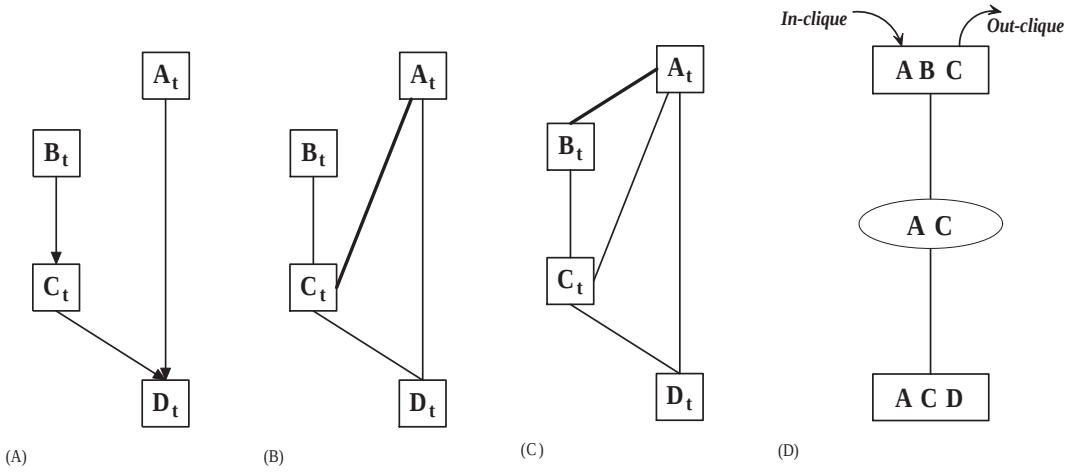


Figure 2.21: Construction of the Junction Tree JT_1 associated to the initial slice BN_1 of the DBN shown in figure 2.20: BN_1 (A); moralisation of BN_1 (B); ensuring that the outgoing interface I_1 is a clique (C); and the resulting Junction Tree (D).

incoming interface composed by 3 nodes: A_t, B_t and D_t. Therefore, in order to minimise the overall computational cost, the interface algorithm is usually applied to the outgoing interface rather than the incoming one.

Two modified junction trees are constructed from the 2TBN model: a junction tree JT_1 associated to the first time-slice BN_1; and a generic JT_t associated to the 1.5DBN. Note that the 1.5DBN model, also known as the "one and a half slice" model, contains all the nodes from the 2TBN second slice, but only the outgoing interface I_t from the first slice of the 2TBN. For example the 1.5DBN associated to the model in figure 2.20 contains all the nodes apart from D_t.

Considering the initial time-slice BN_1 (left slice of the 2TBN taken out of context), the first Junction Tree JT_1 can be obtained using a slightly modified version of the algorithm outlined in section 2.3. For example, the initial junction tree JT_1 associated to the 2TBN model in figure 2.20 can be obtained:

- extracting the right slice of the 2TBN as shown in figure 2.21(A)
- moralising the graph as in figure 2.21(B)
- ensuring that outgoing interface I_1 (A_t, B_t and C_t) is fully connected, making it a clique through the insertion of some additional arcs as in figure 2.21(C)

- triangulating the graph of figure 2.21(C) and forming the junction tree JT_1 in figure 2.21(D)

- marking the clique containing the outgoing interface I_1 both as the in-clique and as the out-clique.

The resulting junction tree is then treated as a static BN: potentials associated to JT_1 are initialised to the unity and then multiplied by the Conditional Probability Densities (prior probabilities).

In analogy to JT_1, the generic junction tree JT_t associated to the 1.5DBN model of figure 2.22(A) can be constructed and initialised following these steps:

- moralisation as shown in figure 2.22(B)

- completion of the two outgoing interfaces I_1 (A_t, B_t and C_t) and I_2 (A_{t+1}, B_{t+1} and C_{t+1}), as in figure 2.22(C)

- triangulation, figure 2.22(D)

- generation of a junction tree, figure 2.23(A), selecting the in-clique I_1 and the out-clique I_2 (cliques containing the interface nodes)

- initialisation of all potentials to 1.

The 1.5DBN model contains only the interface I_1 from the left slice of the 2TBN, where I_1 dynamically represents the execution of the inference process since its beginning (inference on JT_1). Since the 1.5DBN is intended to perform inference exclusively on the right slice of the 2TBN, evidence is applied only to the right slice nodes, potentials multiplied only by the 2TBN right slice CPDs, and probability queried only on nodes with a time index $t+1$.

All junction trees JT_t from $t = 1$ to $t = T$ are connected together through their in-clique and out-clique interfaces. Inference is then performed in each tree separately, and messages are passed forwards and backwards between adjacent junctions trees via their interface cliques. Note that the expensive DAG triangulation algorithm needs to be applied to the generic 2TBN only twice (in order to obtain JT_1 and JT_t), the resulting JTs are then rubber stamped for the desired number of frames connecting them through their interface nodes. For example, given the DBN of figure 2.20

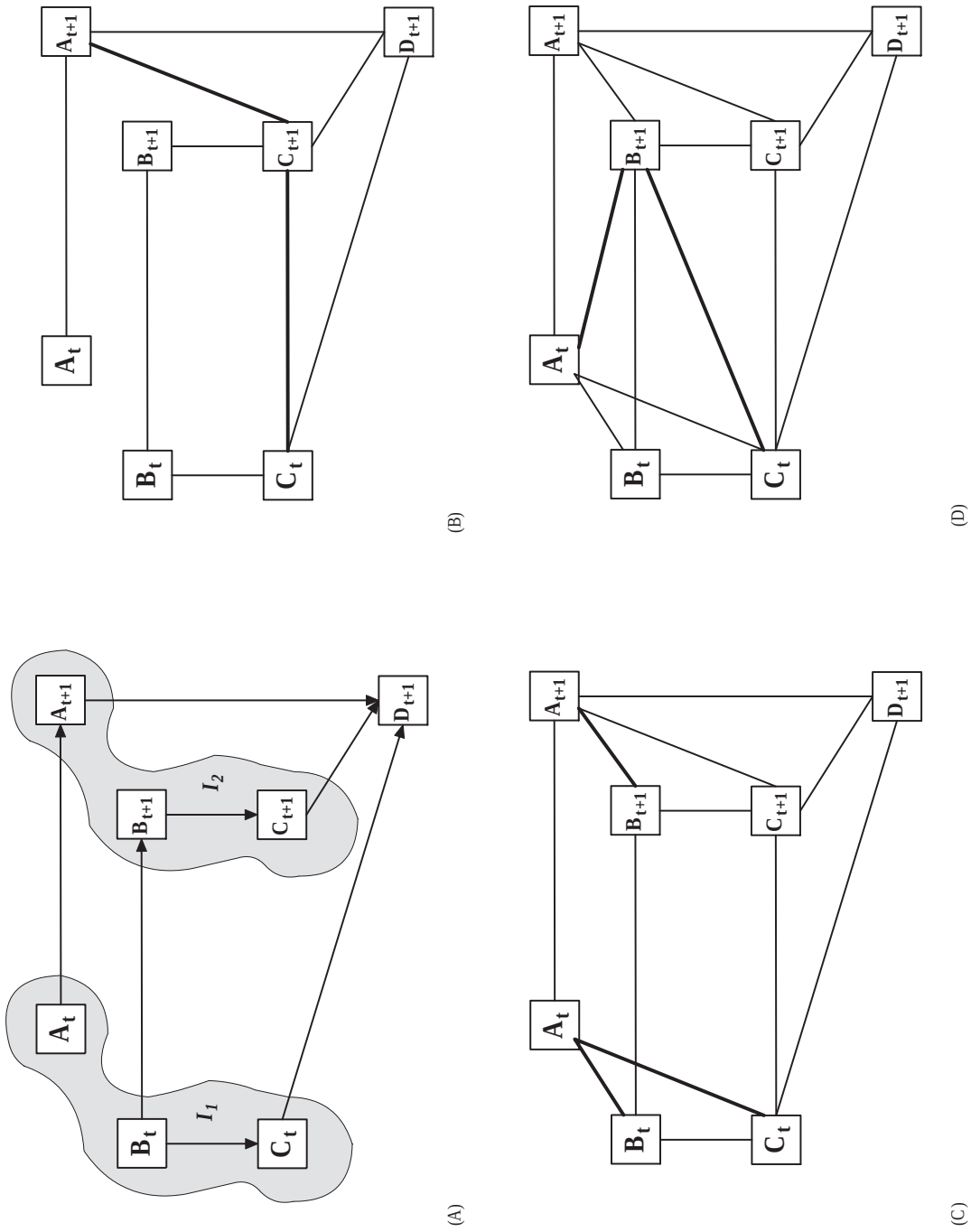


Figure 2.22: Building of the generic Junction Tree JT_t associated to the 2TBN in figure 2.20: the 1.5DBN model (A); moralisation (B); ensuring that the outgoing interfaces I_1 and I_2 are fully connected (C); and triangulation (D).

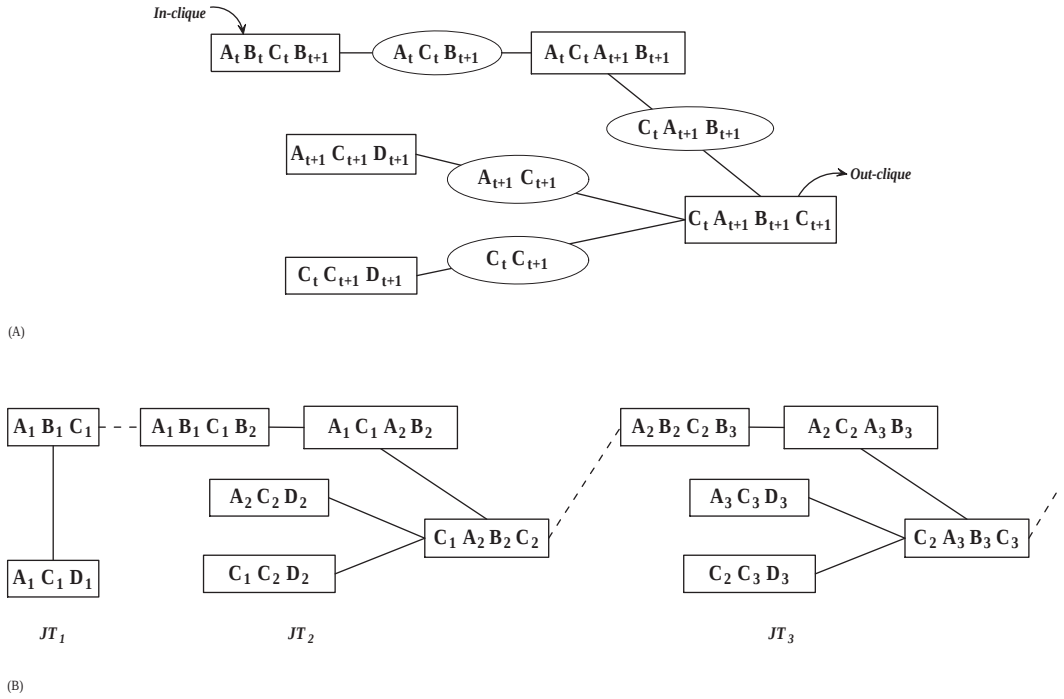


Figure 2.23: Generic Junction Tree JT_t (A), and the unrolled set of Junction Trees JT_1, JT_2, ..., JT_T associated to model in figure 2.20 (B).

and the associated junction trees JT_1 and JT_t shown in figure 2.21(D) and 2.23(A) respectively, the unrolled network of JTs can be constructed as in figure 2.23(B). The inference computation starts from the first slice $t = 1$ and the initial junction tree JT_1. The beliefs of JT_1 are firstly updated, and the auxiliary variable α_1 obtained by marginalising the out-clique potential to the outgoing interface: variables not in the interface are summed out. Since the out-clique of JT_1 exactly corresponds to the outgoing interface, the estimation of α_1 is reduced to:

$$\alpha_1 = \phi_{A_1B_1C_1} \quad . \tag{2.46}$$

Inference is then performed on the second junction tree JT_2, the auxiliary variable α_1 is multiplied onto the in-clique potential:

$$\phi_{A_1B_1C_1B_2} = \alpha_1 \cdot \phi_{A_1B_1C_1B_2} \tag{2.47}$$

updating JT_2 potential $\phi_{A_1B_1C_1B_2}$. The beliefs of JT_2 are then propagated using the static BN message passing algorithm (section 2.3.5), and α_2 estimated marginalis-

ing $\phi_{C_1A_2B_2C_2}$:

$$\alpha_2 = \sum_{C_1} \phi_{C_1A_2B_2C_2} \quad . \tag{2.48}$$

Time is incremented once again and the third junction tree JT_3 is taken into account by applying α_2 to the in-clique potential:

$$\phi_{A_2B_2C_2B_3} = \alpha_2 \cdot \phi_{A_2B_2C_2B_3} \quad . \tag{2.49}$$

This procedure is then iterated for all the following slices, until the last frame $t = T$ has been reached and processed through JT_T. Note that the auxiliary variables α_t derived from the out-clique potentials are the only quantities that need to be propagated: inference on JT_t needs α_{t-1} from the previous frame as input (together with the current observations) and provides α_t as the output. Therefore given α_{t-1} each junction tree JT_t is completely independent from the surrounding slices.

Although the interface algorithm, operating on independent time slices and generalising the process to a single generic JT, provides a significant improvement in terms of computational overhead over naïve DBN flattening, performing exact inference on DBNs with a large state-space might be very slow. However it is possible to adopt approximate inference approaches such as: the factorised frontier algorithm (Murphy and Weiss, 2001), the Boyen-Koller algorithm (Boyen and Koller, 1998), loopy belief propagation (Pearl, 1988) generalised for DBNs, or various sampling approaches. Murphy (2002a) provides a comprehensive and rich compendium of approximate inference algorithms for DBNs.

For example the Boyen and Koller (1998) approach approximates the interface algorithm by partitioning the outgoing interface into multiple disjoint sets. These node clusters are assumed to be independent, reaching the lowest approximation errors when node clusters are chosen so that arcs do not cross cluster boundaries [15]. The reduced computational cost derives from the factorisation of the out-clique potentials, and is achieved by replacing potentially large joints with a product of simpler terms (e.g. assuming two clusters $\{A\}$ and $\{B,C\}$ and factorising $\alpha_1 = \phi_{A_1B_1C_1} \approx \sigma_{11} \cdot \sigma_{12} = \phi_{A_1} \cdot \phi_{B_1C_1}$). This approximation results in having multiple separate "in and out cliques" which need to be marginalised and propagated independently.

[15] Children are included in the same cluster of their parents, for example nodes B_t and C_t in figure 2.20.

2.7 Software packages

Having a set of algorithms to perform inference efficiently and independently of the particular BN or DBN topology, has made possible to develop a unified set of tools to perform standard activities such as model parameter training, state space decoding, and sampling. Unfortunately the advantage of having more freedom with the model structure is counterbalanced by a higher demand in terms of computational resources (both memory and execution time). Considerable effort has been concentrated in developing faster and memory parsimonious algorithms and tools.

BN models can be easily developed thanks to the availability of a large number of open source and commercial toolkits: from Kevin Murphy's Bayesian Network Toolbox [16] to the Microsoft Bayesian Network Editor and Toolkit (Kadie et al., 2001). However the situation is quite different for Dynamic Bayesian Networks. At the time of writing only two toolkits are available to develop DBN based approaches: the Intel Probabilistic Networks Library (PNL) and the Graphical Model ToolKit (GMTK) (Bilmes and Zweig, 2002).

The Intel Probabilistic Networks Library is an open source library [17] principally targeted on DBNs with some additional support for undirected graphical models. Graphical models and their parameter sets are formally defined using conventional programming languages (C/C++) rather than introducing an ad-hoc formal language (as in GMTK). However PNL can be easily interfaced to R and wrappers for Matlab are also available. PNL supports both exact and approximate inference, offering an implementation of the Junction Tree algorithm for exact inference and loopy belief propagation for approximate inference. PNL has been employed in several research works: Liu et al. (2005) proposed a parallel implementation of Module Networks (an extension of BNs characterised by large set of variables with similar dependencies); Wang et al. (2004) investigated a DBN approach for face tracking combining multiple visual features; Portinale et al. (2007) implemented a reliability analysis tool to convert Dynamic Fault Trees into DBNs; and Huttenhower and Troyanskaya (2006) investigated a Bayesian approach to predict protein functions from genomic data. This library although targeted on academic research also allows to quickly implement graphical model based end-user applications (Portinale et al.,

[16] Available from: `http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html`
[17] Available from: `http://sourceforge.net/projects/openpnl/`

2007), or to integrate the probabilistic framework within pre-existing applications. The latest release of PNL is version 1.0 (2005).

The Graphical Model ToolKit is a standalone research toolkit principally focused toward experimenting with new models. GMTK provides a specialised formal language to describe DBNs, supporting also advanced features such as switching parent nodes (Bayesian multinets), and an efficient infrastructure to optimise the number of Gaussians required by GMMs. Although Gaussian mixtures with a basic diagonal covariance vector represent the default; full, sparse, and banded diagonal covariance matrices are supported as well. Since the toolkit has been initially targeted toward Automatic Speech Recognition (Bilmes, 2003; Bilmes and Bartels, 2005), n-gram language models and Factored Language Models (Bilmes and Kirchhoff, 2003) are supported natively. Differently from PNL undirected graphs are not supported and the source code is not publicly available. However this project is under constant development (new tools, new features, speed improvements, etc.) and its use is actively supported. A particular attention has been dedicated toward a computationally efficient implementation, also including distributed computing features such as parallel training.

Both toolkits (PNL and GMTK) have been implemented in C++ for better computation performances, and both support a wide range of conditional probability distributions, such as full and sparse conditional probability tables, conditional Gaussians, deterministic relationships, and decision trees. Note that neither of them supports continuous hidden variables, thus all hidden nodes should be discrete. Instead, observable variables can be both continuous or discrete.

All the DBN related experiments reported in this thesis have been performed using GMTK. The adoption of this toolkit rather than PNL is motivated by several practical aspects. The DBN specification language offered by GMTK is simple and effective, allowing to reduce the model development time. Native support of Switching DBNs and Factored Language Models was greatly appreciated during the development of the dialogue act recogniser (chapter 7). Moreover GMTK offers a good trade-off between simplicity and computational efficiency. Last but not least GMTK is a research tool under active development.

2.8 Motivation

Graphical models are a powerful tool for modelling complex stochastic processes, providing thus a unifying foundation for a wide range of AI related tasks such as: speech processing, computer vision, information retrieval, data mining, gene sequencing, cognitive modelling, fault diagnosis and industrial process control. In particular DBNs provide a common set of building blocks which can be used to describe several classic models (from Gaussian mixture models and Kalman filters to hidden Markov models and their latest extensions) or to develop radically new approaches. DBNs offer a practical way to represent multiple hidden random variables encoding conditional probability assumptions among these variables. Moreover DBN (and graphical models in general) offer several advantages over basic hidden Markov models:

- increased flexibility in the state-space factorisation and structuring thanks to an arbitrary set of hidden variables X_t^i;
- feature-space factorisation by defining an arbitrary set of observable nodes Y_t^j;
- capability to integrate some problem specific knowledge into the model, and therefore ability to develop potentially more discriminative models;
- improved and more parsimonious use of the parameter space;
- unified graphical-mathematical formalism.

DBNs provide not only a modular and intuitive graphical representation, but also a strong common mathematical formalism/background. This graphical infrastructure is able to describe both simpler models such as GMMs and HMMs, or richer models including coupled HMMs, factorial HMMs, hierarchical HMMs, and semi-Markov models (Smyth et al., 1997; Bilmes, 2003). This formalism is not only a container and an efficient representation for such well-known models, but provides a good starting point for the development of innovative model structures.

The Junction Tree algorithm offers a general purpose framework to perform probabilistic inference over Bayesian Networks. Approaches such as the interface algorithm extend the JT framework to DBNs, allowing to efficiently estimate exact

probabilistic inference on DBNs (irrespectively of the implemented model). Since inference is the fundamental step both for model parameter learning and for model decoding, in the last decade two general purpose software packages to perform training and testing of DBN models have been developed. In particular GMTK allows to quickly implement, train, and test models from a virtually unlimited set of DBN topologies: ranging from basic classical approaches (GMMs, HMMs, etc.) to novel DBN based models (chapters 4 and 7) .

Chapter 3

Multimodal meeting recordings and feature extraction

3.1 Introduction

We are interested in the automatic recognition of human-human interactions in the context of multiparty meetings. Our goal is to automatically structure meetings both in terms of "group meeting actions" and "dialogue acts". These represent the same communicative process, a multiparty conversation, employing two different levels of abstraction. Meeting actions provide a coarse representation of the meeting structure representing whole group interactions such as discussions, monologues, and presentations. Dialogue acts (DAs) focus on individual meeting participants highlighting their communicative intentions. DAs can be interpreted as the atomic blocks of a conversation, often including categories such as statements, questions, offers, and suggestions.

These two tasks can be addressed using a common probabilistic framework based on two steps, extraction of relevant features from the audio-visual recordings, followed by feature integration and modelling. The first step is based on the application of signal processing techniques to the raw meetings recordings. The latter one is implemented by the mean of a statistical model, which can be used to relate high level categories (i.e: meeting actions or dialogue acts) to complex patterns in the observed feature set. In the previous chapter we have introduced Dynamic Bayesian Networks, the general-purpose graphical modelling infrastructure

that we will adopt in all our meeting action (chapter 5) and dialogue act recognition experiments (chapter 7).

All the modelling approaches proposed in this thesis, being based on supervised learning, require significant amounts of manually annotated examples in order to be trained. This chapter outlines the annotated data resources (section 3.2) which were adopted in our experiments: the M4 (section 3.2.1), ICSI (section 3.2.2) and AMI (section 3.2.3) meeting corpora, along with the Switchboard and Fisher corpora (section 3.2.4). The M4 meetings, being annotated in terms of group meeting actions, formed the basis for our meeting action recognition experiments of chapter 5. Joint dialogue act recognition was performed (chapter 7) both on ICSI and AMI meeting data using their respective DA annotation schemes.

The second part of this chapter (section 3.3) introduces the four feature families which were adopted in our experiments: prosodic (section 3.3.1), "speaker turn" (section 3.3.2), lexical (section 3.3.3), and visual features (section 3.3.4). They act as an interface between audio-video signals and DBN probabilistic models, representing in a compact way the information contained in the low level recordings.

The last section of this chapter (section 3.4) summarises the two feature sets which were employed respectively for the meeting action (section 3.4.1) and for the dialogue act (section 3.4.2) recognition tasks.

3.2 Annotated resources and annotation schemes

Supervised probabilistic models, like the DBN approaches introduced in the previous chapter, need collections of manually annotated examples for their training. In this section we will present the data resources that have been adopted on that purpose.

The M4 meeting corpus (section 3.2.1), which is annotated in terms of group meeting actions, provides an ideal training and testing environment for the meeting action recognition experiments reported in chapter 5. Similarly dialogue act recognition experiments were performed on the ICSI (section 3.2.2) and AMI (section 3.2.3) corpora: two large meeting collections annotated in terms of dialogue acts. Finally some additional data resources, useful to enhance statistical language modelling (section 7.5.3), will be outlined in section 3.2.4.

3.2.1 The M4 meeting corpus

This multimodal data collection consists of 69 short meetings, recorded at IDIAP as part of the M4 (MultiModal Meeting Manager) European Union IST project, referred to as the M4 Meeting Corpus (McCowan et al., 2003) [1]. Each recording in the corpus captures the interaction of four participants following an overall meeting structure that was planned in advance. The structure is defined in terms of a sequence of meeting actions from the dictionary outlined in chapter 5: monologue (per speaker), discussion, note-taking, presentation, and presentation at whiteboard. *Monologues* focus on individual meeting participants performing a prolonged oration addressing the group. *Discussions* consist in multiparty conversations involving two or more participants. During *note taking* all participants take some time to write down notes about the meeting. *Presentations* are similar to monologues, except that the main speaker makes use of slides to accompany his oral presentation. During *white-board presentations* the orator makes use of a white-board to illustrate concepts and facilitate his presentation. The resultant meetings thus follow a high level "script", but the individual participant behaviours and language are unscripted and natural. This predefined segmentation of the meeting constitutes a "perfect" ground truth reference annotation in terms of meeting phases. Although the sequence of meeting phases strictly obeys to a predefined structure, the transitions between different phases are very natural and follow the given timeline with a certain approximation. Therefore the boundaries between meeting phases tend to be smooth and spread over several seconds. The communicative phases "dissolve" one into another, making infeasible, even for a human observer, to pinpoint the exact instant when the transition had taken place. These meeting actions may be considered both as group social actions and as meetings phases (McCowan et al., 2003), and used to segment meetings identifying different communicative phases. The meeting action sequence provides a description of the meeting structure, and builds up a simple semantic language, which may be used to formulate queries for a retrieval system, or to assist meeting browsing.

The corpus consists of more than five hours of synchronised multichannel audio-video recordings. Recordings took place in an instrumented meeting room (figure 3.2). Each participant wore a wired lapel microphone, and a 8 element circular mi-

[1] This corpus is publicly available from: `http://mmm.idiap.ch/`

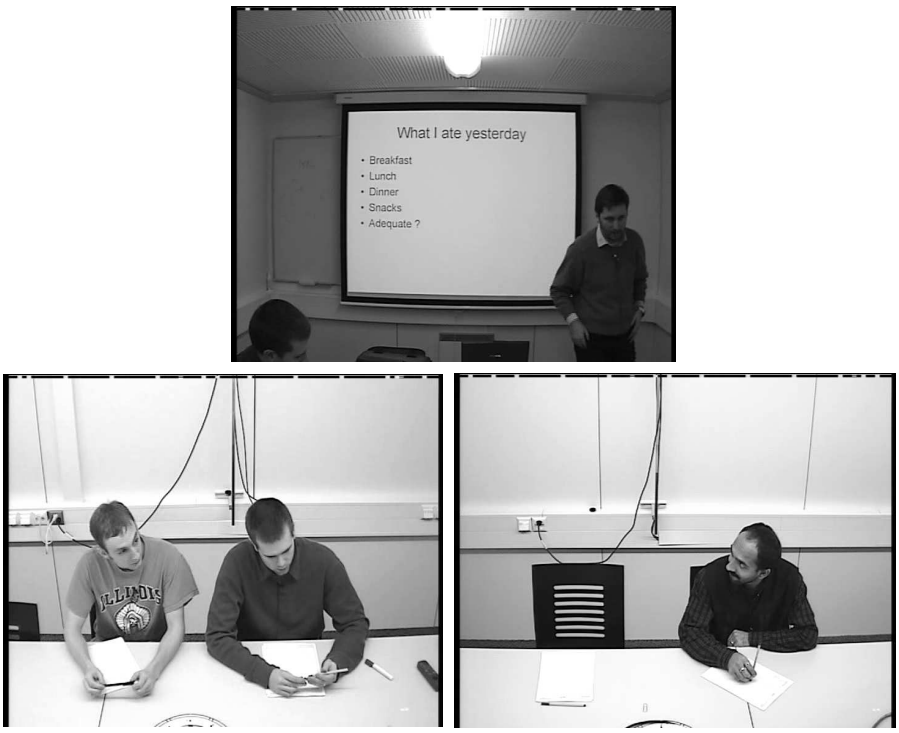


Figure 3.1: A meeting scene example captured with 3 fixed video-cameras: white-board and projector screen region (top image) and two opposite sides of the table (bottom images).

crophone array was placed on the table between participants. Note that nothing was done to prevent reverberation or to reduce environmental noise, thus offering realistic recording conditions. Orthographic (word-level) transcriptions were provided for 30 of the 69 meetings. Three fixed cameras provided visual recordings of the meeting activity (figure 3.1). Two wall mounted cameras gave a landscape view of each side of the table (usually two people in shot). The third camera framed the projector screen and the white-board area. As for audio, the video recording conditions were unconstrained with phenomena such as object occlusions and changes in illumination.

3.2.2 The ICSI meeting corpus

The ICSI meetings corpus (Janin et al., 2003) consists of 75 naturally occurring research group meetings at the International Computer Science Institute in Berkeley during the years 2000–2002, recorded using wireless close-talking microphones

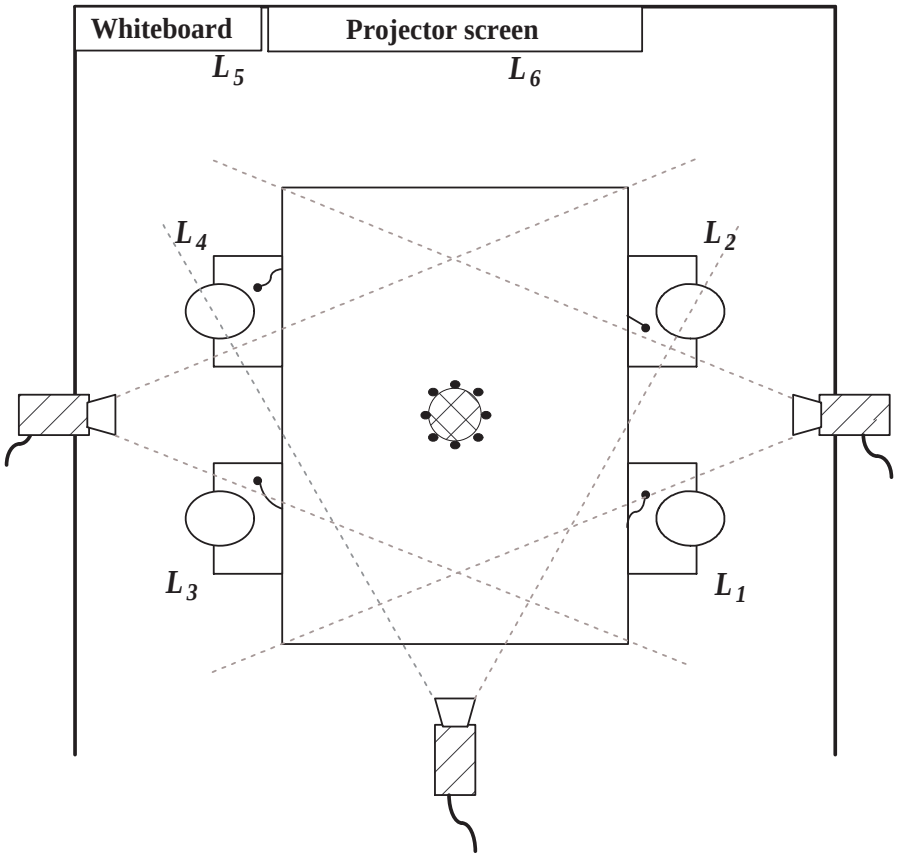


Figure 3.2: Layout of the M4 instrumented meeting room equipped with: 3 fixed video-cameras, 4 wired lapel microphones, and a 8 element circular microphone array.

worn by each participant (in addition, there were also four tabletop microphones). Each meeting lasts about one hour and involves an average of six participants, resulting in about 72 hours of multichannel audio data. The corpus contains human-to-human interactions recorded from naturally occurring meetings. Moreover, having different meeting topics and meeting types, the data set is heterogeneous both in terms of content and structure.

Orthographic transcriptions are available for the entire corpus, and each meeting has been manually segmented and annotated in terms of Dialogue Acts, using the ICSI MRDA scheme (Shriberg et al., 2004) shown in table 3.1 and extensively discussed in the MRDA annotation manual (Dhillon et al., 2004). The MRDA scheme is based on a hierarchy of DA types and sub-types (11 generic tags and 39 specific sub-tags), and allows multiple sub-categorisations for a single DA unit. This

Statement	
s	**Statement**
Questions	
qy	**Yes/No Question**
qw	**Wh-Question**
qr	**Or Question**
qrr	**Or Clause After Y/N Question**
qo	**Open-ended Question**
qh	**Rhetorical Question**
Floor Management	
fg	**Floor Grabber**
fh	**Floor Holder**
h	**Hold**
Backchannels	
b	**Backchannel**
bk	Acknowledgement
ba	Assessment/Appreciation
bh	Rhetorical Question Backchannel
Responses	
aa	Accept
aap	Partial Accept
na	Affirmative Answer
ar	Reject
arp	Partial Reject
nd	Dispreferred Answer
ng	Negative Answer
am	Maybe
no	No Knowledge
Action Motivators	
co	Command
cs	Suggestion
cc	Commitment
Checks	
f	Follow Me
br	Repetition Request
bu	Understanding Check
Restated Information	
r	Repeat
m	Mimic
bs	Summary
bc	Correct Misspeaking
bsc	Self-Correct Misspeaking

Supportive Functions	
df	Defending/Explanation
e	Elaboration
2	Collaborative Completion
Politeness Mechanisms	
bd	Downplayer
by	Sympathy
fa	Apology
ft	Thanks
fw	Welcome
Further Descriptions	
fe	Exclamation
t	About-Task
tc	Topic Change
j	Joke
t1	Self Talk
t3	Third Party Talk
d	Declarative Question
g	Tag Question
rt	Rising Tone
Disruptions	
%	*Indecipherable*
%-	*Interrupted*
%–	*Abandoned*
x	*Nonspeech*
Nonlabeled	
z	Nonlabeled

Table 3.1: MRDA labels used for the annotation of the ICSI meeting corpus: **generic tags**, specific tags and *disruptions*. Source: the MRDA annotation manual (Dhillon et al., 2004).

extremely rich annotation scheme results in more than a thousand unique DAs, although many are observed infrequently. To reduce the number of sparsely observed categories, we have adopted a reduced set of five broad DA categories (Ang et al., 2005; Zimmermann et al., 2006a). Unique DAs were grouped into five generic categories: statements, questions, backchannels, fillers and disruptions. A set of conversion rules was applied to map the MRDA scheme to the 5 broad DA labels based scheme. Table 3.2 outlines the ICSI official conversion scheme used in our experiments ("classmap 01b" in the corpus documentation); different conversion schemes are feasible but will lead to different results. On this class-map Shriberg et al. (2004) reported a good overall inter-annotator agreement Kappa (Cohen, 1960; Carletta, 1996) of about $k = 0.8$. The Kappa statistic measures the agreement achieved among annotators beyond chance: $k = 1$ suggest absolute agreement among the annotations and $k = 0$ full disagreement.

The distribution of these 5 broad DA categories across the corpus is shown in table 3.3. Statements are the most frequently occurring unit, and also the longest, having an average length of 2.3 seconds (9 words). All the other categories (except backchannels which have an average duration of 0.14 seconds) share an average length of 1.6 seconds (6 words). An average meeting contains about 1 500 DA units.

In order to have directly comparable results a formal subdivision into three data sets has been proposed by Ang et al. (2005): a training set of 51 meetings (about 80 000 Dialogue Act units), a development set of 11 meetings (13 500 DAs) and a test set of 11 meetings (15 000 DAs). This leaves out 2 of the 75 meetings (transcriber meetings *Btr001* and *Btr002*), which were excluded because of their different nature (Zimmermann et al., 2006b). All our DA segmentation and classification experiments were conducted on the proposed dataset subdivision.

3.2.3 The AMI meeting corpus

The AMI meeting corpus (Carletta et al., 2006) is a multimodal collection of annotated meeting recordings. It consists of about 100 hours of meetings collected in three instrumented meeting rooms. About two thirds of the corpus consists of meetings elicited using a scenario in which four meeting participants, playing different roles on a team, take a product development project from beginning to completion.

Source MRDA label	Target DA category	Examples
|s	Statement	fgˆtc|sˆcs , fg|sˆaapˆdf
s*	Statement	sˆbkˆrt , sˆftˆt3
br*	Statement	brˆrt
*%	Disruption	sˆftˆt3.%
*%-	Disruption	qhˆcs.%-
*%–	Disruption	qo.%–
*x	Disruption	qr.x
b*	Backchannel	bˆtc
f*	Filler	fg , fh
h*	Filler	h
|q	Question	qwˆdˆeˆg
q*	Question	qhˆbr

Table 3.2: Rule based mapping from MRDA labels to five broad DA categories

Category	% of total DA units	% of corpus length
Statement	58.2	74.5
Disruption	12.9	10.1
Backchannel	12.3	0.9
Filler	10.3	8.7
Question	6.2	5.8

Table 3.3: Distribution of DA categories by percentage of the total number of DA units and by percentage of corpus length.

Category	DA class	Proportion %
Information exchange	*inform*	26.6
	elicit inform	3.4
Individual or group action	*suggest*	7.5
	offer	1.2
	elicit offer or suggestion	0.5
Comment on previous discussion	*assess*	16.7
	elicit assessment	1.7
	comment about understanding	1.8
	elicit comment understanding	0.2
Social function	*be positive*	1.8
	be negative	0.1
No speaker intention	*backchannel*	17.6
	stall	6.3
	fragment	13.0
Other	*other*	1.8

Table 3.4: The six broad categories and fifteen specialised Dialogue Act classes used in the AMI corpus DA annotation scheme, with the percentage of DAs in each class.

The scenario portion of the corpus consists of a number of meeting series, with four meeting per series. Each series of four meetings involves the same four participant roles (project manager, marketing expert, industrial designer, and user interface designer), and comprises project kick-off, functional design, conceptual design, and detailed design meetings. The remaining meetings in the corpus, “non-scenario”, are naturally occurring meetings, with 3–5 participants.

The aim of the corpus collection was to obtain a multimodal record of the complete communicative interaction between the meeting participants. To this end, the meeting rooms were instrumented with a set of synchronised recording devices, including wireless lapel and headset microphones for each participant, an 8-element circular microphone array, six video cameras (four close-up and two room-view), capture devices for the whiteboard and data projector, and digital pens to capture

the handwritten notes of each participant. The corpus was manually annotated at several levels, including orthographic transcriptions, various linguistic phenomena including Dialogue Acts, head and hand movements, and focus of attention [2]. The DA annotation scheme for the AMI corpus, outlined in table 3.4, is based around a categorisation tailored for group decision making, and consists of six broad categories and a total of 15 DA classes. Each DA unit is assigned to a single class, corresponding to the speaker's intent for the utterance. In order to reduce the uncertainty during the DA annotation process, the three human annotators involved in this task strictly adhered to the recommendations formulated in the AMI DA Annotation Guidelines (2005). Moreover a small portion of the data was annotated by all the 3 annotators to check the reliability of the scheme, and reannotated towards the end of their involvement with the DA annotation process, to assess the stability in their judgements. The inter-annotator agreement according to the Kappa statistics (Cohen, 1960; Carletta, 1996) was found to be in the range $k = 0.83 - 0.89$. The distribution of the DA classes, shown in table 3.4, is rather imbalanced, with over 60% of DAs corresponding to one of the three most frequent classes (inform, backchannel or assess). Over half the DA classes account for less than 10% of the observed DAs. An example of the reference DA annotation using the 15 DA classes is shown in table 3.6: manually transcribed utterances are first segmented and then labelled with individual DA tags. This annotation scheme is different to the one used for the ICSI corpus (section 3.2.2), thus it is not possible to test a DA recognition system developed on the AMI data on the ICSI corpus or vice-versa.

We performed our DA recognition experiments on the 138 meetings that form the scenario subset of the AMI corpus, following the subdivision into training, development, and test sets suggested in the corpus documentation (table 3.5). The scenario meetings are organised in 35 series of (normally) four meetings: 25 series of meetings have been assigned to the training set, five to the development and five to the test set (table 3.5).

[2] The annotated corpus is freely available from: `http://corpus.amiproject.org/`

Subset	Meetings	#meetings	#series
Training set	ES2002, ES2005-2010, ES2012-2016 IS1000-1007 TS3005 TS3008-3012	98	25
Development set	ES2003, ES2011, IS1008, TS3004, TS3006	20	5
Evaluation set	ES2004, ES2014, IS1009, TS3003, TS3007	20	5
All scenario data		138	35

Table 3.5: The subdivision of the AMI scenario data into training, development and evaluation sets.

3.2.4 Additional annotated data resources

The SWITCHBOARD corpus (Godfrey et al., 1992) consists of about 2 500 orthographically transcribed telephone conversations by 500 unique speakers. More than 250 hours of unconstrained conversational speech were recorded at Texas Instruments using a fully automatic recording infrastructure. Conversational topics were automatically chosen and suggested before each conversation. The whole collection of conversations has been manually transcribed and aligned at word level, resulting in nearly 3 millions of words. More than 200 000 utterances and 1.4 millions of transcribed words have been annotated in terms of Dialogue Acts using the very rich SWBD-DAMSL (Jurafsky et al., 1997b) annotation scheme comprising 226 unique tags, or 42 clustered DA labels.

The Fisher corpus (Cieri et al., 2004) consists of more than 16 000 English telephone conversations on a wide range of elicited topics, resulting in about 2 000 hours of recorded speech, which were orthographically transcribed.

Although it is not possible to use these corpora directly as training data for tasks such as meeting action recognition or DA recognition (both using the AMI or the ICSI annotation schemes), they represent valuable additional sources of transcribed conversational data. The Fisher corpus is of particular utility, since it contains over 10 million words, making it an order of magnitude larger than the AMI and ICSI corpora.

Start time (seconds)	End time (seconds)	Participant ID	DA label	Utterance (reference orthographic transcription)
775.33	776.97	B	[Inform]	We're a bit behind
776.12	777.67	D	[Elicit-Assessment]	Do you know what I mean
777.32	777.89	C	[Backchannel]	Yeah
777.67	779.25	D	[Fragment]	Like so sort of like how do you
777.70	778.02	A	[Backchannel]	Yeah
779.25	786.49	D	[Inform]	I I mean one one way of looking at it would be well the people producing television sets maybe they have to buy remote controls
786.49	793.63	D	[Inform]	Or another way is maybe people who have T.V. sets are really fed up with their remote control and they really want a better one or something
792.52	793.22	C	[Fragment]	I know um
794.52	801.38	C	[Inform]	My parents went out and bought um remote controls because um they got fed up of having four or five different remote controls for each things the house
794.53	794.97	D	[Fragment]	But
799.88	801.75	D	[Assess]	Right Right
801.38	804.33	C	[Inform]	So um for them it was just how many devices control
801.75	802.20	D	[Fragment]	Okay so
804.47	804.72	D	[Assess]	Right
804.72	810.39	D	[Inform]	so in function one of the priorities might be to combine as many uses
806.46	806.98	B	[Backchannel]	Yeah
811.51	815.56	B	[Elicit-Assessment]	Right so do you think that should be like a main design aim of our remote control d you know
814.98	815.81	D	[Assess]	I think so
815.56	820.83	B	[Elicit-Assessment]	do your your satellite and your regular telly and your V.C.R. and everything
815.81	817.04	D	[Backchannel]	Yeah yeah
818.45	818.67	D	[Backchannel]	Yeah
819.79	821.05	D	[Stall]	Well like um
821.05	828.28	D	[Inform]	maybe what we could use is a sort of like a example of a successful other piece technology is palm palm pilots
828.28	834.09	D	[Inform]	They're gone from being just like little sort of scribble boards to cameras M.P. three players telephones
832.76	833.20	B	[Backchannel]	Mm-hmm
835.22	836.31	D	[Inform]	everything agenda
836.31	840.68	D	[Inform]	So like I wonder if we might add something new to the to the remote control market
840.25	840.69	B	[Assess]	Yeah
840.68	844.06	D	[Inform]	such as the lighting in your house or um
843.99	847.62	B	[Inform]	Or even like you know notes about um what you wanna watch
847.62	850.33	B	[Inform]	Like you might put in there oh I want to watch such and such and look a
849.08	849.78	D	[Backchannel]	Yeah
849.78	851.06	D	[Backchannel]	yeah
850.33	851.55	B	[Assess]	Oh that's a good idea
851.06	851.42	D	[Fragment]	An
851.55	853.03	B	[Inform]	So extra functionalities
852.79	853.54	D	[Assess]	Yeah
853.54	862.9	D	[Inform]	Like p personally for me at home I've I've combined the um the audio video of my television set and my D.V.D. player and my C.D. player
862.9	864.62	D	[Inform]	So they w all work actually function together
864.62	866.91	D	[Inform]	but I have different remote controls for each of them
866.59	867.24	B	[Backchannel]	Mm-hmm
866.91	870.27	D	[Assess]	So it's sort of ironic that that then they're in there
872.03	872.56	D	[Stall]	um
874.02	876.45	D	[Inform]	you know the sound and everything it's just one system
876.45	877.77	D	[Inform]	But each one's got its own little
876.88	877.46	B	[Backchannel]	Hmm
878.95	879.22	D	[Inform]	part

Table 3.6: Reference orthographic transcription and manually annotated Dialogue Act units from the AMI corpus (scenario meeting ES2002a).

3.2.5 Discussion

The annotated data resources outlined in this chapter, although inspired by the same underlying goal (i.e. collecting multiparty conversational speech), present several major differences.

For example the M4 corpus shows natural interactions but is severely constrained in terms of topics and conversation lengths. Moreover only lapel microphone recordings have been collected, increasing all the technical difficulties related to automatic speech recognition and speech processing.

The ICSI corpus contains a large selection of occasional planning and update meetings, held by participants who know well each other and have previously discussed similar or related topics. The resulting conversations are so natural, unconstrained and rich in over-specialised topics that sometimes they are inaccessible to a naïve external listener. Moreover the ICSI corpus lacks of video recordings and audio recordings are loosely-synchronised.

The AMI corpus attempts to address all these issues including two distinct recording sets, i.e. fully unconstrained and scenario elicited meetings, and synchronised multichannel audio-visual recordings. However the amount of collected data is limited if compared to the CTS Fisher corpus and some of the hi-level annotations are available just for the scenario subset. The collection of a new corpus is a complex process involving a large effort in terms of: work, organisation, and resources; thus setting some compromises is unavoidable.

3.3 Feature extraction and post-processing

Most of the meeting recordings outlined in the previous sections involve audio and eventually video, but the communicative process is spread between several modalities including speech, prosody, gestures, handwriting, and facial and body expressions. Further streams of data could be captured easily: for example handwriting could be recorded through whiteboard capturing devices, graphic tablets or digital pen/paper; and this was done during the AMI meeting data collection (section 3.2.3). Unfortunately this is not the case for modalities such as gestures or facial expression, for which the use of specialised recording devices is impractical and invasive. When specialised recordings are not available, it is possible to extract

multiple modalities from single streams. For example, speech could be separated from noise and other sound sources using microphone array beamforming, physical motion could be measured using image processing techniques, and it may be further integrated into a gesture recogniser. Note that this is a simplified view of the problem, because a single modality corresponds to multiple different streams: for example, speech is manifested not only as a sound but also as a lip motion. The situation is further complicated if we consider the correlations that exist between different modalities, such as speech and gestures (McNeill and Duncan, 2000). The analysis of natural human communication based on multiple streams corresponding to recordings of different modalities is a difficult task, since acoustic recordings are corrupted by environmental noise and room reverberations, video recordings include occlusions and environmental changes, the participant interactions are highly spontaneous and usually unconstrained, and there is a very wide range of topics, speakers, speaking styles and accents.

In this section we present four feature families related to prosody (section 3.3.1), turn taking dynamics (section 3.3.2), lexical content (section 3.3.3), and visual level of motion (section 3.3.4). Prosodic features (F0, energy, and rate-of-speech) can be directly extracted from the raw audio recordings. Speaker turn features rely on microphone array processing hence they are indirectly based on the audio recordings as well. Similarly the orthographic transcription needed by the lexical features can be obtained automatically from the audio streams [3]. The first three feature families are based on speech and audio communicative modalities because these are predominant in meetings. However video recordings have been exploited in order to estimate the visual motion level of several relevant areas (i.e.: participants head and hand regions).

3.3.1 Prosodic features

Different features related to the speech modality can be extracted for each meeting participant, using audio recordings provided by individual headset or lapel microphones. In our experiments we focused on five prosody related features: a smoothed estimate of the fundamental frequency (F0), an estimate of the syllabic

[3] Automatic transcriptions are available for the ICSI and AMI meeting corpora (section 7.3.0.1) but not for the M4 corpus.

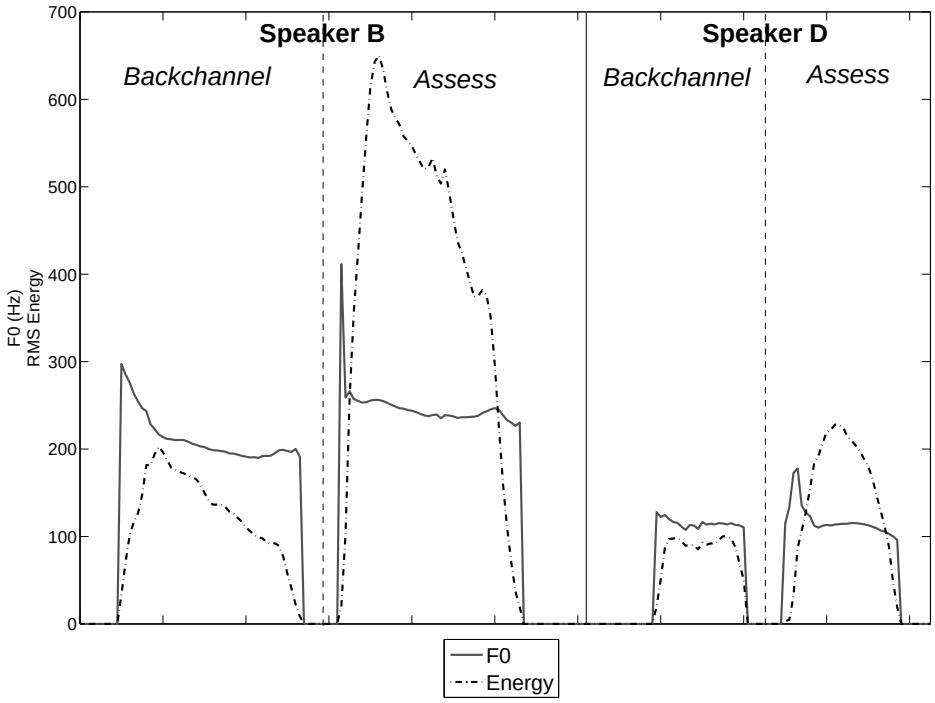


Figure 3.3: Pitch and energy features associated to the same word in two different contextual situations (*backchannel* and *assess* Dialogue Acts).

rate of speech, speech signal energy, word length and pause duration. The resulting prosodic features aim to capture variations in speaking style, highlighting specific aspects of the speech modality. For example in a Dialogue Act (DA) classification task the function of the word "yeah" as a *backchannel* or *assess* DA unit can be disambiguated from its prosody. An example of this is shown in figure 3.3 where the different use of the same word is clearly associated to specific F0 and energy patterns.

Moreover prosodic features highlight when a particular speaker is active (as shown in figure 3.4) eventually suggesting the level of engagement in the conversation.

3.3.1.1 Denoised pitch estimation

A rough estimate of the intonation contour can be obtained from the raw audio recordings by adopting a pitch tracking algorithm (Mousset et al., 1996). However the resulting F0 estimates are affected by errors (Murray, 2001; Khurshid and Den-

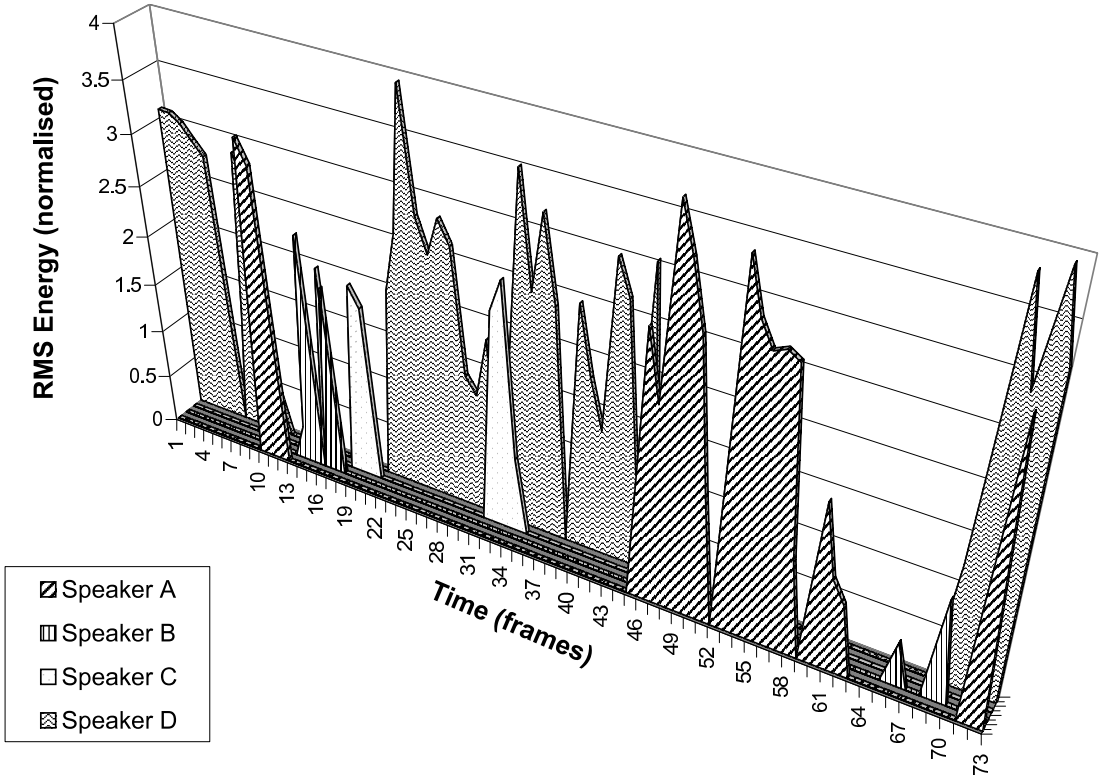


Figure 3.4: RMS energy features associated to four speakers being involved in a *discussion* (M4 meeting corpus).

ham, 2004) leading to inaccuracies. The top of figure 3.5 shows the automatically extracted pitch contour (dotted lines) for a short excerpt taken from the M4 corpus. Although the presence of background noise is very limited, non-vocal noises and cross-talk from the other meeting participants often cause spurious peaks and noisy F0 estimates. However these artifacts can be reduced or even removed by filtering the estimated pitch track.

The smoothed and denoised F0 is estimated in two steps: an initial F0 contour estimate using the ESPS get_f0 pitch tracking algorithm (Talkin, 1995) [4], followed by a chain of three filters, inspired by Sonmez et al. (1998), that denoise the initial estimate of F0. The filter chain used for that purpose is shown in figure 3.5. A histogram filter removes incorrect estimates arising from other undesired sound sources, followed by a median filter to smooth the F0 contour removing spurious peaks, and a linear interpolation filter that provides a piecewise continuous

[4] Available from: `http://www.speech.kth.se/snack/`

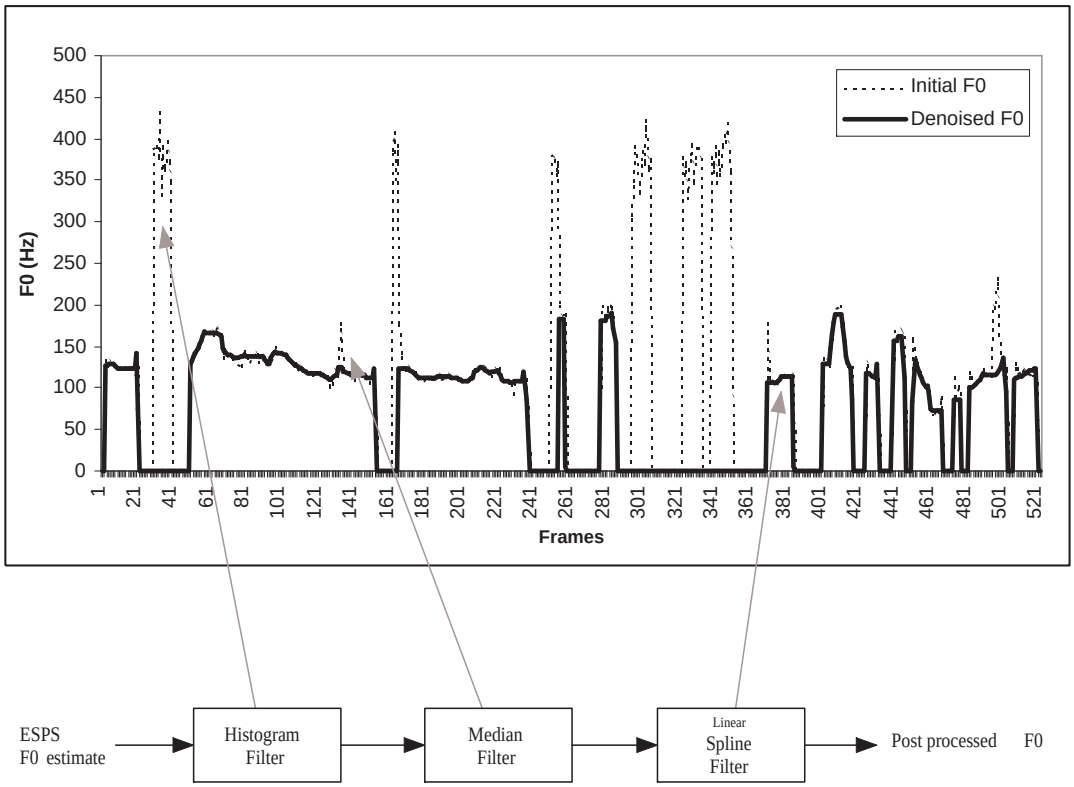


Figure 3.5: Smoothing and denoising of the F0 estimate.

smoothed output. The average F0 for the current channel [5] is estimated using the entire meeting. The average F0 is then used to normalise the instantaneous pitch estimates, in order to have comparable features for different speakers.

3.3.1.2 Multichannel energy estimation

Speech energy can be easily estimated measuring the root mean square signal energy of the recorded audio samples. Having multiple microphone channels, this procedure can be applied to each individual audio channel. However the estimated energies need to be normalised to compensate for different channel gains, making them comparable.

The logarithm of the root mean square energy $E_j(t)$ can be evaluated for each

[5] M4 and AMI meeting recordings are characterised by a tight correspondence between audio channels and speakers: meeting participants are not allowed to share or swap their microphones.

microphone channel j:

$$E_j(t) = \log\left(\sqrt{\frac{1}{N}\sum_{n=0}^{N}\left(x_j(n+\lfloor t\cdot f_{fea}\rfloor\cdot N)\right)^2}\right) \qquad N=\frac{f_s}{f_{fea}} \tag{3.1}$$

where $x_j(n)$ represents the n^{th} audio sample from the microphone channel j, f_s is the audio recordings sampling frequency, and N represents the number of audio samples accounted for by each feature frame. Note that the features are extracted at a lower sampling rate f_{fea} than the original waveform, for example assuming a sampling frequency f_s of 16 KHz and a feature vector frame length of 16 milliseconds ($f_{fea} =$ 62.5 Hz), each feature sample is based on $N = 256$ audio samples. $E_j(t)$ is then normalized as follows (Pfau et al., 2001):

$$E_{norm,j}(t) = E_j(t) - E_{min,j} - \frac{1}{M}\sum_{m=1}^{M} E_m(t) \quad . \tag{3.2}$$

The minimum log-energy $E_{min,j}$ can be interpreted as an estimate of the noise floor level recorded by channel j. Therefore it needs to be subtracted in order to compensate for different channel gains. The last term represents the mean log-energy averaged across all $M = 4$ channels. We are primarily interested in sounds (speech) that occur only in proximity of the channel. Considering one channel m at a time, those sounds should be considerably above the background noise (multi-channel averaged energy).

Note that since headset microphones are more directional than lapel microphones and closer to the sound source, the resulting recordings have an improved signal-to-noise ratio and are less affected by cross talk between adjacent speakers. Therefore the last normalisation term of equation 3.2 can be omitted during the headset based energy estimation.

3.3.1.3 Syllabic rate of speech

Rate of speech can be estimated using two methods: using the phone/word level segmentation provided by an Automatic Speech Recognition system, or directly from the waveform. The first method assumes that the rate of speech is inversely proportional to the automatically [6] estimated word durations (section 3.3.1.4). However when a reliable automatic transcription is not available, as for the M4 meeting

[6] A more accurate estimate can be obtained aligning the reference orthographic transcription to the waveform through forced alignment.

corpus, the syllabic speaking rate can be estimated from the acoustic signal using the algorithm *mrate* (Morgan and Fosler-Lussier, 1998). The *mrate* approach integrates the output of multiple rate of speech estimators by averaging their individual estimates. Three acoustic based estimators are integrated in Morgan and Fosler-Lussier (1998): a peak counting algorithm applied to the wide-band energy envelope, a sub-band version of the peak counting approach applied to the average product over all pairs of compressed sub-band energy trajectories, and the *enrate* estimator (Morgan et al., 1997). *Enrate* is based on the first spectral moment of the wideband energy envelope computed over few seconds.

In order to have comparable features for different speakers, speech rate estimates are normalised across the entire meeting dividing them by the average rate of speech for a given speaker.

3.3.1.4 Time and duration related features

Both the ICSI and AMI meeting corpora have been automatically transcribed (section 7.3.0.1) [7], thus information about word boundaries can be exploited to estimate inter-word pauses and word durations.

Interword pauses are estimated using word boundary times obtained from aligning the automatic transcription with the acoustic signal, and re-scaled in order to have a unitary range. Note that long pauses between words may highlight sentence boundaries and thus be a strong cue to DA segmentation (chapter 7).

Similarly the word length can be estimated as the word duration normalised by the mean duration for that word computed on the entire dataset. Therefore the word length is inversely proportional to the rate of speech, neglecting estimation errors.

3.3.2 Speaker turn features

Face-to-face meetings display a complex turn-taking structure. The dynamics of this process can be extremely useful to distinguish between different meeting phases (chapter 4) in the context of the M4 meeting action recognition experiments (chapter 5). For example, during dialogues speakers tend to alternate frequently, speaking for shorter periods.

[7] The speech recogniser's output supplies both the sequence of recognised words and their starting and ending time.

To investigate the turn-taking process, it is necessary to detect speech activity for each participant in the meeting. This is difficult using the lapel microphone signals for two reasons. Firstly, since they are wired microphones, M4 meeting participants only wear the lapel microphones while seated, which makes them impossible to use when someone is presenting a talk or standing at the whiteboard. Secondly the lapel microphones are omnidirectional and it is difficult to distinguish whether a signal is the speech of the participant wearing the microphone, or crosstalk from another speaker (Pfau et al., 2001; Wrigley et al., 2005). Instead, we used microphone array recordings to detect speaker activity. More generally in the AMI project (Carletta et al., 2006) the microphone array is regarded as the primary recording condition (NIST, 2004). Although the "speaker turn features" outlined in this section were designed for meeting action recognition (section 3.4.1), being thus specific to the M4 meeting recordings, they can be adapted to different corpora and novel tasks.

3.3.2.1 Sound source localisation

A microphone array can be regarded as a steerable directional microphone, but, compared with an orientable microphone, there are no moving parts. The steering direction can be imposed at any time during or after the recording session using a beamforming process. It is therefore possible to steer the virtual microphone in any direction, evaluating sound activity at a specific spatial location (Lathoud and McCowan, 2003). In the M4 meetings there are only six spatial regions in which participants spent most of their time (regions L_1, ...,L_6 in figure 3.2): the four seating regions that are individually associated with participants, the white-board and a presentation space near the projection screen. We detected continuous sound activities $L_i(t)$ in each of these six regions i, which were used as a basis for features to describe the turn-taking process. Each $L_i(t)$ is directly proportional to the probability of observing an active sound source (a meeting participant speaking or generating noise) in the spatial region i at time t, and it is zero when no activity is detected.

3.3.2.2 Turn taking detection

We constructed a 216-element feature vector to describe the turn-taking process at each time. The vector S consists of all 6^3 possible products of the 6 sound activity

locations $L(t)$ during a time window of 3 frames (Dielmann and Renals, 2004a):

$$S_{ijk}(t) = L_i(t) \cdot L_j(t-1) \cdot L_k(t-2) \;\; \forall i,j,k \in [1,6]$$

where each vector $S_{ijk}(t)$ highlights the turn taking interaction pattern around the time t. Considering, for simplicity, a smaller turn taking matrix evaluated only on two frames:

$$S_{ij}(t) = L_i(t) \cdot L_j(t-1) \;\; \forall i,j \in [1,6]$$

the diagonal elements $S_{ii}(t)$ highlight whether a speaker L_i active at time $t-1$, is still speaking at time t. The terms above the diagonal (S_{ij} , $i < j$) are greater than zero when it is likely that L_i is speaking after L_j. Similarly $S_{ij} > 0$, $i > j$ implies that L_j at time $t-1$ and L_i at time t are both active. When all: S_{ii},S_{jj},S_{ij} and S_{ji} are greater than zero, it is likely that a discussion (turn-taking alternation) between L_i and L_j is taking place. A similar discussion applies to $S_{ijk}(t)$.

Dimension reduction of S_{ijk} using principal component analysis was not effective, with reductions below 200 dimensions resulting in a degradation in performance. Thus we used the unreduced 216-element feature vector in our meeting action recognition experiments.

3.3.3 Lexical features

Lexical information embedded into textual transcriptions can be employed to extract relevant cues from the current conversation. Two lexically related features are proposed in this section: a monologue/dialogue discriminator (section 3.3.3.1) and a word informativeness indicator (section 3.3.3.2). The monologue/dialogue discriminator is strictly related to the M4 group meeting action recognition task (chapter 5). A more sophisticated approach based on the adoption of factored language models for Dialogue Act classification (section 7.5.1) was applied to the ICSI and AMI DA recognition tasks (chapter 7).

3.3.3.1 Lexical style/genre discrimination

Monologues and dialogues are characterised by different speaking styles and different language models. In particular we hypothesise that the distribution over words is different for transcripts from these two meeting phases. Using a transcript for

each speaker we constructed trigram language models for each communicative context that we wish to recognise. In our "meeting action" recognition experiments (chapter 5) we estimated language models for monologue and discussions only, but the idea could be extended to more elaborate domains. Note that the "dialogue act" recogniser outlined in chapter 7, adopting a probabilistic language model to discriminate between multiple DA categories, stems from the same intuition outlined here. However on the DA recognition task the language models (section 7.5) were integrated within the recognition framework rather than employed to generate stand-alone features.

The approach is illustrated in figure 3.6. Trigram language models correspond to monologues (M_1) and discussions (M_2). Those multinomial distributions over words are estimated using transcriptions from the training data set, and then used to partition unseen word sequences from the test set. Note that the language models are estimated employing all the transcribed words, irrespectively of the function they serve in the discourse. For example words such as "yeah", "it", "is", "think" are more frequent in a discussion, and terms such as "this", "these", "he", "what" occur more frequently during monologues. Each word w_t (together with its context w_{t-1}, w_{t-2}, if available) contained in the transcription under test is compared with both the models M_1 and M_2 ($K = 1,2$) and assigned to the class with the highest probability $P_{LM}(w_t \mid w_{t-1}, w_{t-2}; M_k)$:

$$\tilde{k}(w_t) = \arg\max_{k \in K} \{P_{LM}(w_t \mid w_{t-1}, w_{t-2}; M_k)\}$$

where $\tilde{k}(w_t)$ is the output of the classifier.

The resultant sequence of output symbols is noisy, with $\tilde{k}(w_t)$ constantly switching between the two states (small dots of figure 3.7). However if we consider the symbol density, the output is much more stable (lines of figure 3.7). Therefore we smooth the output by evaluating the relative frequency of $\tilde{k}(w_t)$ over a sliding window of 24 words. This window length has been arbitrarily chosen, but it seems not to be critical because values between 20 and 30 are equally acceptable. This lexically-based approach is able to classify unseen word sequences as monologues or discussions with a percentage of correctly classified words of about 93% [8] .

[8] Average recognition using leave-one-out cross-validation strategy on 30 manually transcribed short meetings from the M4 meeting corpus.

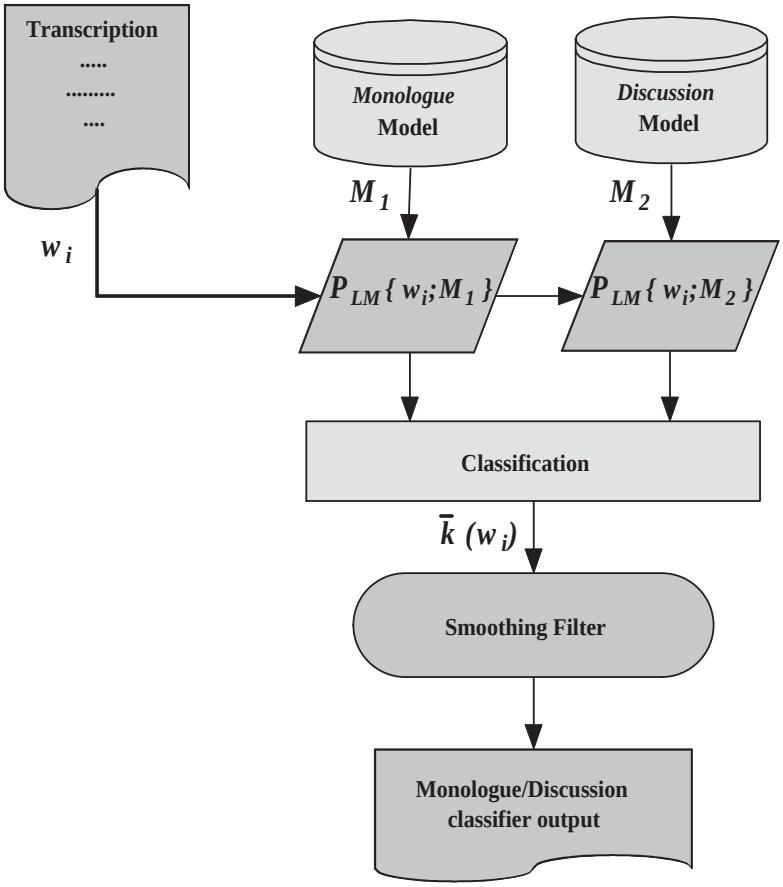


Figure 3.6: Overview of the “monologue/discussion” classifier.

Removing the smoothing step and considering the noisy sequence $\tilde{k}(w_t)$ the classification accuracy falls to 78%. A lower bound on the class accuracy of 48% is obtainable by drawing the symbols by chance, according to the prior distribution.

3.3.3.2 Word informativeness

The word informativeness was computed to be the ratio between local term frequency within the current conversation and absolute term frequency across the whole meetings collection. Terms which are more relevant for the current meeting will assume scores well above the unity. For example: “stability”, “strongest”, “spherical”, “soda”, “ball-shaped”, etc. are the most informative words of the AMI scenario meeting ES2016a.

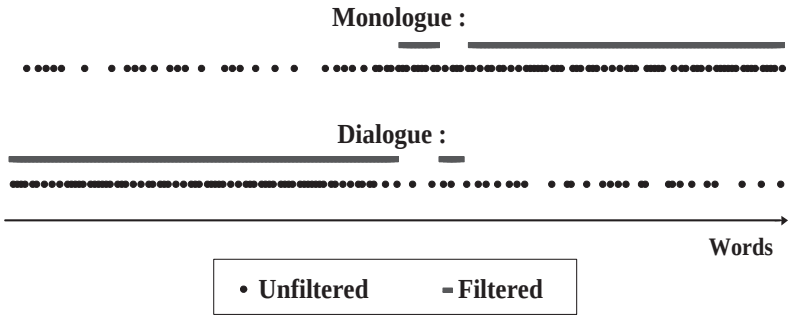


Figure 3.7: Filtering of $\tilde{k}(w_t)$.

3.3.4 Visual features

Meetings provide a well defined and highly constrained environment for video and image processing. Participants spend most of the time in a few spatial locations—they move location rarely and there are relatively few physical actions. In the case of the M4 corpus (section 3.2.1), cameras are fixed, most furniture does not move and lighting conditions are partially constrained. However, participants are free to perform any action or gesture and do whatever they like. Therefore object occlusions are relatively frequent, and nothing has been done to facilitate object tracking (i.e.: there is no "blue screen" or preassigned colors for clothing or furniture). Note that exposure settings are different for each camera. In particular this is a critical issue for the camera oriented on the bright projection screen and dark white-board area (figure 3.1).

Although the M4 recordings were made using high quality equipment and good video resolutions (full frame PAL), regions of interest represent only a small fraction of the entire scene, providing a relatively low resolution. This resolution is sufficient for tasks such as tracking the head, hands, and other objects of a similar size. Close-up video recording will be required to address problems such as lip feature extraction for audio-video speech recognition or eye-gaze tracking for conversational attention prediction (Vertegaal et al., 2001).

We assume visual information about the participants is correlated with meeting phases. For example, a speaker who is highly involved in the conversation will tend to gesticulate, and the use of a white-board involves a complex sequence of physical actions, such as standing up, walking and writing. Under that assumption

we are interested in extracting a set of region based low level visual activities (Basu et al., 2001), that could improve the recognition of highly visual actions such as note taking and presentations.

3.3.4.1 Optical flow based motion estimation

Our efforts are concentrated on the two cameras oriented towards the meeting table. Each of those captures a scene with two speakers. As mentioned above, meeting participants spend most of the time sitting, therefore only the upper body part is visible through those cameras (figure 3.1). In each scene we analyse four areas: the head and hand regions for each of the two participants in shot. Instead of recognising and tracking head and hand blobs (Zhang et al., 2004a) we have chosen a faster and more flexible approach that does not require an appearance model or any form of (re-)initialisation.

Our system relies on an optical flow based algorithm, which is used to track a fixed number ($n = 100$) of feature points. We have adopted an enhanced version of the "Kanade Lucas Tomasi" (KLT) feature tracker outlined in Shi and Tomasi (1994). In particular the condition used to select the tracking feature set was revised and extended. Given a sequence of images $F(x,y,t)$ which includes a slowly moving object (region Ω surrounding the point (x,y)), it is possible to assume that the brightness of the object does not change given two adjacent frames:

$$F(x+dx,y+dy,t+dt) \approx F(x,y,t) \quad . \tag{3.3}$$

Approximating $F(x,y,t)$ around the point (x,y) with a first order Taylor's series:

$$F(x+dx,y+dy,t+dt) \approx F(x,y,t) + (\nabla F(x,y,t))^T \cdot \begin{pmatrix} dx \\ dy \end{pmatrix} + \frac{\partial F(x,y,t)}{\partial t} \cdot dt \tag{3.4}$$

the optical flow constraint equation can be derived combining equations 3.3 and 3.4:

$$-\frac{\partial F(x,y,t)}{\partial t} = (\nabla F(x,y,t))^T \cdot \begin{pmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{pmatrix} \tag{3.5}$$

where $-\frac{\partial F(x,y,t)}{\partial t}$ represents the brightness variation speed, $\nabla F(x,y,t)$ the spatial gradient of the image brightness, and $(\frac{dx}{dt}\ \frac{dy}{dt})^T$ the object's speed. A unique solution,

given a weight function $W(x,y)$, can be found minimizing the following least square error equation in a neighbourhood Ω of (x,y):

$$\sum_{(x,y)\in\Omega} W(x,y)^2 \left[(\nabla F(x,y,t))^T \cdot \begin{pmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{pmatrix} + \frac{\partial F(x,y,t)}{\partial t} \right]^2 \tag{3.6}$$

and thus solving the following system:

$$\sum_{(x,y)\in\Omega} W(x,y)^2 \cdot A \cdot \begin{pmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{pmatrix} = \sum_{(x,y)\in\Omega} W(x,y)^2 \cdot \begin{pmatrix} \frac{\partial F(x,y,t)}{\partial x} \\ \frac{\partial F(x,y,t)}{\partial y} \end{pmatrix} \cdot \frac{\partial F(x,y,t)}{\partial t} \tag{3.7}$$

where

$$A = \nabla F(\Omega,t)^T \cdot \nabla F(\Omega,t) = \begin{pmatrix} \left(\frac{\partial F(\Omega,t)}{\partial x}\right)^2 & \frac{\partial F(\Omega,t)}{\partial x} \cdot \frac{\partial F(\Omega,t)}{\partial y} \\ \frac{\partial F(\Omega,t)}{\partial x} \cdot \frac{\partial F(\Omega,t)}{\partial y} & \left(\frac{\partial F(\Omega,t)}{\partial y}\right)^2 \end{pmatrix} \quad . \tag{3.8}$$

Assuming that $det\left(\sum_{(x,y)\in\Omega}\left[W(x,y)^2 \cdot A\right]\right) \neq 0$ the solution of KLT optical flow problem is given by:

$$\begin{pmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{pmatrix} = \sum_{(x,y)\in\Omega} \left[W(x,y)^2 \cdot A\right]^{-1} \cdot W(x,y)^2 \cdot \begin{pmatrix} \frac{\partial F(x,y,t)}{\partial x} \\ \frac{\partial F(x,y,t)}{\partial y} \end{pmatrix} \cdot \frac{\partial F(x,y,t)}{\partial t} \quad . \tag{3.9}$$

If λ_1, λ_2 are the two eigenvalues of A, the system will be well conditioned if the smallest eigenvalue λ_{min} is large enough:

$$\lambda_{min} = min(\lambda_1, \lambda_2) >> 0 \tag{3.10}$$

Adopting this condition, Shi and Tomasi (1994) stated that "good features are the ones that can be tracked well", proposing therefore to track feature regions with a particularly rich texture.

Being interested in tracking skin-like regions, we have extended the feature quality metric (3.10) proposed by Shi and Tomasi with an additional condition over the candidate region's color:

$$P(\Omega \mid Skin) \geq P_{th} = 0.5 \quad . \tag{3.11}$$

It is thus feasible to evaluate the chromatic distribution of skin blobs (Yang et al., 1998), and to use that distribution to estimate the probability of a given region Ω

to be skin. The chromatic space can be represented through different bases: here we adopted the luminance and chrominance space $\{Y,Cr,Cb\}$. Skin-like colours are well clustered under the $\{Cr,Cb\}$ subspace, and a 3 component Gaussian Mixture Model (GMM) was trained using unseen skin blobs. The resulting skin color model provided an easy way to estimate the skin probability (equation 3.11) for each candidate region Ω. Therefore a good feature is now one that can be tracked well (equation 3.10) and has a high probability to be part of a skin area (equation 3.11).

Our approach to the video feature extraction process is depicted in figure 3.8. Each video stream is processed on a frame-to-frame basis, the skin probability is estimated and used to select and track 100 features, i.e. the image regions with a rich texture. Those features are processed off-line. Feature trajectories that are too long and have a limited amount of motion (often associated to red objects in the background) are automatically removed. The next step consists of partitioning the trajectory space into four regions (head and hand areas for the two participants in shot). Note that both hands of a participant are included into a single region. Four Gaussian distributions (one centroid for each region) were estimated off-line using the entire video sequence. This rough global estimation was then refined on a frame basis, by using a k-means clustering: the initial assignation provided the 4 Gaussian centroids provides the initialisation for the k-means clustering. If a trajectory T of length n is assigned to a set $K(i),1 = 1,..,n$ of different regions, and $\tilde{K}$ is the most frequent assignment, then the whole trajectory T is classified as part of region $\tilde{K}$. The proposed system is based on an enhanced implementation of the KLT algorithm provided by the OpenCV (2001): GMM skin model, k-means clustering, and its Gaussian initialisation were developed ad-hoc.

For each frame, and for each region, two video features were extracted: the average feature motion intensity, and the approximate motion direction. Thus from each raw video signal an 8-element feature vector is extracted each frame (i.e. two features, two regions, two participants). The feature vectors from the two cameras are combined, resulting in a 16-element global video feature vector. Owing to the recording conditions of the third camera (projector screen and whiteboard area), motion vectors extracted from this source are less reliable, and were excluded from our experimental setup.

This approach exploits few assumptions about the scene structure without pre-

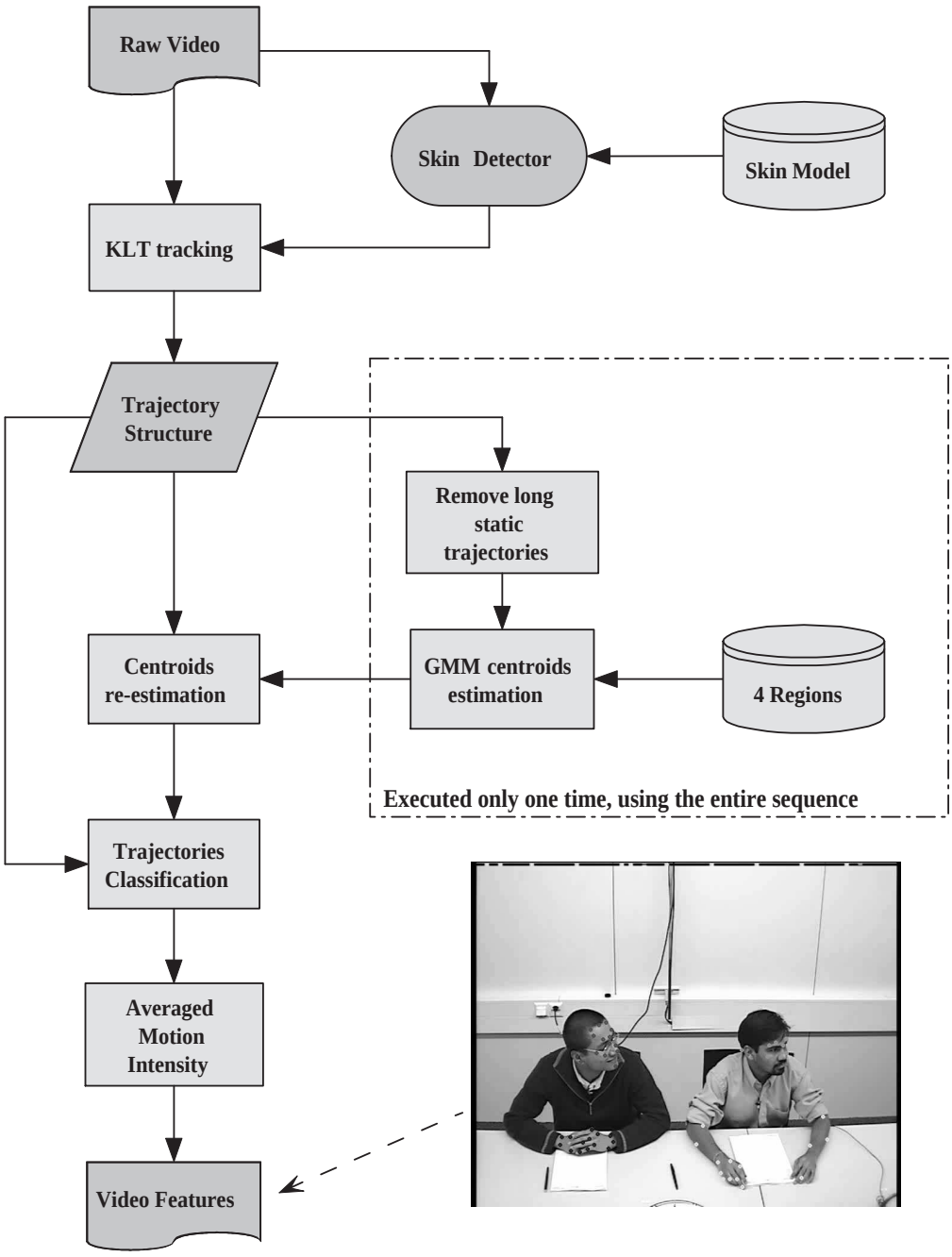


Figure 3.8: Overview of the “video features” extraction process.

Features	Meeting action recognition	Dialogue act recognition
	M4 corpus	ICSI and AMI corpora
Pitch	✓	✓
Syllabic rate of speech	✓	
RMS Energy	✓	✓
Word length		✓
Interword pause duration		✓
Speaker turns	✓	
Lexical features	✓	(FLMs)
Word informativeness		✓
Visual features	✓	

Table 3.7: Features used for the meeting action recognition (on the M4 meeting corpus) and the dialogue act recognition tasks (both on ICSI and AMI data).

tending to precisely identify head or hand blobs. Therefore object occlusions are only partially handled. However, recovering from an unexpected event is fast and completely automatic, there is no need for manual initialisation, and this technique translates well between domains. The system is able to operate in the presence of complex coloured backgrounds without any performance degradation, and is able to cope with gradual illumination changes.

3.4 Tasks and feature setups

All the feature families presented in the previous sections highlight different aspects of the rich human-to-human interactions which compose a multiparty meeting. The aim is to combine these complementary views, in order to automatically detect complex communicative events such as meeting actions, or to structure the underlying conversation in terms of dialogue acts. However some feature families were de-

veloped specifically for just one of the two tasks, for example speaker turns are intimately related to group meeting actions. Table 3.7 outlines the feature selection that were adopted for the meeting action recognition task (chapter 5) and for the dialogue act recognition experiments (chapter 7).

3.4.1 Meeting action recognition

The meeting action recognition task consists of segmenting meetings into a sequence of group meeting actions such as monologue, discussion and presentation. A set of six time-continuous feature families were extracted every 0.5 seconds from the audio-visual recordings, and processed using the multistream DBN infrastructure outlined in chapter 5.

Three prosodic features (F0, syllabic rate of speech, and energy) were extracted from each of the four lapel recordings (table 3.7). Note that word length and pause duration would have required the accurate timing information from the ASR output, which unfortunately is not available for the M4 meeting corpus. However, in order to improve the quality of F0 and rate of speech, discretised versions of the speaker activities estimated using microphone array processing techniques (section 3.3.2) were used to mask inactive lapel microphone channels. Unfortunately prosodic features could not be extracted when participants were presenting a talk or standing at the whiteboard, since the use of wired lapel microphones is feasible only when participants were close to the table. Therefore the prosodic feature set is partially incomplete and also affected by estimation errors.

Information about the overall turn taking dynamics was included through a 216 dimensional speaker turn feature vector (section 3.3.2). The lexical based conversation style classifier outlined in section 3.3.3.1 provided a strong cue to discriminate between monologue and dialogue meeting actions, and further evidence to discriminate between different meeting phases is provided by the 16-element video feature vector.

3.4.2 Dialogue act recognition

During the dialogue act recognition process a multiparty conversation is segmented and classified according to a dictionary of mutually exclusive DA labels (the ICSI

and AMI DA annotation schemes are outlined in section 3.2.2 and 3.2.3 respectively). Since the segmentation is based on the orthographic transcription of the conversation, a single feature vector is estimated for each transcribed word. For example the average pitch and energy is estimated for each word using timing from the automatic speech recogniser. Note that accurate word temporal boundaries allow to restrict the F0/energy estimation only to speech segments, excluding noise outbursts and limiting the influence of speaker cross-talk. Moreover averaging F0 on multiple observations within a single word makes the use of pitch smoothing and denoising (section 3.3.1.1) redundant. Word and pause duration features are used in lieu of the syllabic rate of speech, and the turn taking process is modelled through a discourse language model (section 7.4). DA discriminative factored language models (section 7.5) extend the role of the monologue-dialogue classifier used in the meeting action recognition task. Word informativeness was also estimated and included into the resulting 6 dimensional feature vector, composed by: F0 mean and variance, energy, word length, inter-word pauses, and word informativeness.

3.5 Discussion

We are interested in automatically structuring meetings by detecting whole group interactions such as meeting actions, and by highlighting individual meeting participant intentions such as dialogue acts. These two tasks share the same objective and can be interpreted as two different granularities of the same problem.

In this chapter we have outlined the principal meeting data resources which have been adopted in our study. The M4 meeting corpus, being annotated in terms of meeting actions, is ideal for the group meeting action recognition experiments (chapter 5). Similarly the ICSI and AMI meeting corpora form the basis for the dialogue act recognition experiments reported in chapters 7 and 8. Four feature families (section 3.3), related to prosody, speaker localisation, lexical, and visual content, were extracted from the raw audio-video recordings. Features from these 4 families were used to build two distinct feature sets (section 3.4): one focused on group meeting action recognition (section 3.4.1) and one targeted on dialogue act recognition (section 3.4.2).

The two meeting structuring tasks, based on the annotated data and the feature

sets outlined in this chapter and on the DBN modelling framework presented in chapter 2, will be be discussed in the remaining chapters of this thesis. An M4 based meeting action recogniser will be introduced in chapter 4 and discussed in chapter 5. Similarly, automatic dialogue act recognition on ICSI and AMI data will be discussed in chapters 6, 7 and 8.

Chapter 4

Meeting Action recognition

4.1 Introduction

Involvement in meetings is a common experience in daily life, particularly in the workplace, where managers spend more than a day each week in meetings [1]. Meetings perform several functions, such as the resolution of disputes, socialisation, problem solving, planning, or the review of results. Only rarely is a meeting focused on a single task: usually groups are engaged in multiple interdependent functions on multiple concurrent projects (McGrath, 1991).

Traditionally the minutes of a meeting are taken by someone present at the meeting. Unfortunately this is a time consuming job, and often fails to capture all the required information. It would be desirable to have an automatic system to enable efficient organisation, search and recall of the information contained in a meeting, or a set of meetings. Such a system would be required to extract high level information such as meeting phases, meeting tasks, textual transcriptions, topic structure, and summaries(Waibel et al., 2001). These high-level descriptions can provide a multi-perspective analysis of a meeting, more detailed and more objective than a hand-made minute. Moreover such an analysis could facilitate browsing over meeting series, making it possible to search for specific events (Kazman et al., 1996).

This chapter and the following one are concerned with the automatic segmentation of multiparty meetings into a set of predefined group meeting actions or phases: monologue, dialogue, note-taking, presentation, and presentation at the

[1] 3M online survey 1998 (`http://www.3m.com/meetingnetwork/`)

whiteboard. This dictionary of meeting actions (section 3.2.1) represents just one example of the possible points of views under which meetings can be analysed. Nevertheless it provides a useful first step in relating low level multimodal signals (section 3.4.1) to higher level categories.

The recording and analysis of meetings has become a flourishing research area recently, with specific foci including meeting browsing, microphone array processing, speaker tracking and person identification (Schultz et al., 2001; Lee et al., 2002; Janin et al., 2003; Mostefa et al., 2007; Voss and Ehlen, 2007; Renals et al., 2008). Several researchers have focused on the automatic recognition of actions in meetings, at both individual and group levels.

4.2 Individual action recognition

The automatic interpretation of human activities, and the automatic recognition of individual actions in particular application domains, is an active research field. Most of the work in this area relies on a supervised approach: unseen multimodal sequences are interpreted using statistical models estimated using annotated data. Both unimodal and multimodal approaches have been used for such problems.

Hidden Markov models (HMMs) have provided a good framework for unimodal tasks, such as speech or handwriting recognition, and usually form the baseline system for multimodal situations. Starting from the assumption that incorporating more knowledge of the underlying problem into the model can improve the model's accuracy, many HMM variants have been investigated, such as hierarchical HMMs (Fine et al., 1998), coupled HMMs (Brand et al., 1997), buried HMMs (Bilmes, 1999) and semi-Markov models (Ferguson, 1980). An important feature of multimodal analysis is the requirement to process multiple asynchronous and interdependent feature streams. This may be addressed through the use of models based on multiple parallel Markov chains, usually referred as *multistream* models. Oliver and Horvitz (2003), for example, proposed a structured approach to the inference of typical office user activities (e.g.: making a phone call, having a face to face conversation) using features derived from audio and video signals, and computer activity logs. This approach relied on a layered HMM, which is hierarchically composed of multiple HMM chains. At the lowest level there is a signal-analysis HMM which

connects low-level features to an intermediate layer, which forms the observations for a higher level HMM, and so on up to the highest level of the model. Each layer may be trained independently (with a supervised approach) and is characterised by its own temporal granularity.

Multimodal sensing has been used to improve speech-based command and control interfaces (Bolt, 1980; Oviatt, 2003), such as information kiosks or video games. Garg et al. (2003) inferred user presence and focus of attention from low level audio, video and "contextual" features using an ad-hoc developed DBN model. A custom DBN (derived from human expertise) encoded causal relations between multi-modal features (mouth motion, silence detection, skin detector, face detector, etc.) and classes that need to be recognised (visible speaker, frontal view of the speaker, and focus of attention).

Automatic classification of broadcast news is another relevant example of multimodal sensing. For example Snoek et al. (2004) proposed a framework to detect TV news monologues using multiple *style detectors* based on multimodal features (frontal face detector, video optical character recogniser, speech detector and speech recogniser) and a Support Vector Machine based classifier.

Audio-video speech recognition (Potamianos et al., 2003) may be viewed as a particular example of multimodal human activity recognition. This is a well-defined domain and forms a good testing ground for the comparison of different approaches and models. Dupont and Luettin (2000) proposed a synchronised multi-stream hidden Markov model, in which the audio and video streams were processed independently. Partial recognitions were integrated only at particular state space configurations (anchor points). This multistream model was implemented by considering the whole Cartesian product of the two independent stream state spaces (HMMs). Therefore state durations, anchor points, and the amount of synchronism/asynchronism between the streams were all explicitly encoded into the model's state space structure. Another multistream approach, which took advantage of a DBN based formalism, is outlined in Zhang et al. (2003). This approach, using words instead of sub-word units as anchor points, further relaxed the assumption about stream synchronisation. Moreover in this approach, state duration modeling and level of synchronisation between the signals, were implicitly determined.

4.3 Group action recognition

The literature concerning group interaction analysis using multimodal features, is much less developed than that about individual action recognition. Hakeem and Shah (2004) proposed a multi-level structured approach to classify visually-related meeting actions and the meeting genre. Head and hand positions were estimated using a standard condensation tracking algorithm, enhanced with a small set of categorised movement attributes. Sequences of movements were mapped into actions or events by a state machine. A hierarchical set of rules was used to detect higher level meeting activity.

Howard and Jebara (2004) introduced a model for multiple concurrent processes (such as the trajectories of the members of a football team), referred to as a dynamical systems tree. This DBN model consists of a structured hierarchy of aggregating parent Markov chains (aggregating-nodes), and a set of switching linear dynamical systems that are used to discretise the continuous feature space (leaf-nodes). Basu et al. (2001) have investigated the automatic analysis of human interaction in informal settings. Multimodal features (speaker audio activities and motion based visual activities) are related to group behaviours through a coupled HMM (section 2.4.3). Direct computations using such a model, with N chains and Q states per chain, requires NQ^N parameters, making this approach intractable even for small N. Basu et al. approximated the model by taking into account the Q^2 individual interactions between a chain i and neighbouring chains j, instead of considering all the Q^N possible interactions between i and the remaining $N-1$ chains.

There has been some previous work using the same corpus and dictionary of meeting actions that we employ in our meeting action recognition experiments. Note that even sharing the same corpus and the same task, differences in the feature set, the data set subdivision and the evaluation methodology, make a direct comparisons between the experimental results reported in chapter 5 and Reiter and Rigoll (2004); McCowan et al. (2005); Dielmann and Renals (2004a); Zhang et al. (2004a); Dielmann and Renals (2004b) infeasible. An attempt to overcome this situation, by comparing the performance of our DBN multistream model (section 5.4.4) on three different feature setups (IDIAP, Munich and Edinburgh feature sets), can be found in the joint work Al-Hames et al. (2006a). Further details about this comparison

can be found in section 5.5.

Reiter and Rigoll (2004) developed an algorithm to segment meetings in terms of meeting actions, based on a minimum length constraint and dynamic programing. Using automatic speaker segmentation and other hand labeled features, this model was used to classify segments as monologues, discussions, etc. by fusing the output of different basic classification approaches (Bayesian Network, Multilayer Perceptron Network and Radial Basis Network). More recently Al-Hames and Rigoll (2005b) proposed a framework for meeting action classification based on three multimodal features: binary speech and silence segmentation, 4 Mel-frequency cepstral coefficients plus energy, and a visual based global motion vector. These features were modeled using a DBN composed of three partially coupled hidden Markov chains. Experiments applying this DBN approach to artificially perturbed pre-segmented meetings offered improved accuracy compared with a baseline HMM classifier. The latest advancements on the M4 meeting action recognition task, both using clean and corrupted recordings, will be discussed in section 5.6.

McCowan et al. (2005) investigated several approaches to multimodal feature integration and meeting action recognition, investigating both participant and group actions. Both early and late integration approaches were investigated. The best results were achieved with a group-based multistream approach (Dupont and Luettin, 2000), with good results obtained using audio features alone (speaker activity and prosodic features). These results highlighted the fact that although acoustic related features outperform video derived features (such as the positions of head and hands), a multistream approach was essential to achieving good results. This work also employed the asynchronous HMM (Bengio, 2003) to address the task of group action recognition with a model expressly designed to cope with asynchronous multimodal signals.

More recently the same feature families have been modelled using a two-level layered HMM (Zhang et al., 2004a). In this hierarchical approach, features are firstly related to participant actions (such as speaking,writing and idle) through a low-level HMM. A higher level HMM, employing the participant action probabilities and other group level features, is then used to recognise meeting actions. This framework has been adapted to the unsupervised case (Zhang et al., 2004b) in which meetings (or meeting series) are segmented and clustered into a set of hidden meet-

ing actions.

Previously we have outlined a meeting action recognition framework based on acoustic and lexical related features and a layered multistream dynamic Bayesian network model (Dielmann and Renals, 2004a,b). This model combines the advantages of independent feature-stream processing together with a structured approach. In the following chapter we provide a clear and unified view of this framework, proposing some further extensions to the model structure (section 5.4.5).

Chapter 5

DBN models for meeting action recognition

5.1 Introduction

In this chapter we are concerned with the automatic structuring of meetings, based on multistream meeting recordings—primarily audio and video streams captured using multiple microphones and cameras. Analysis of natural human communication based on multiple streams corresponding to recordings of different modalities is a difficult task, since acoustic recordings are corrupted by environmental noise and room reverberations; video recordings include occlusions and environmental changes; the participant interactions are highly spontaneous and usually unconstrained; there is a very wide range of topics, speakers, speaking styles and accents.

We are interested in the recognition of *group meeting actions*, whereby a meeting is interpreted as a sequence of interactions between the participants. Our goal is to segment automatically each recorded meeting into a sequence of group meeting actions. We have used a set of five basic group meeting actions defined by the M4 meeting corpus (section 3.2.1): monologues (per speaker), discussions, note taking, presentations and whiteboard-based presentations (McCowan et al., 2003). *Monologues* are focused on an individual addressing the group, which may provide an active feedback. *Discussions*, in contrast to monologues, involve two or more participants in conversation. *Presentations* are similar to monologues, except that the orator speaks from the projection screen area. Another variant of monologues are

white-board presentations, in which the main speaker makes use of a white-board to explain concepts. Finally, *note taking* is a group action in which participants write down their own notes. These group action symbols are assumed to be mutually exclusive and non-overlapping. Moreover, the meeting action dictionary is also assumed to be exhaustive: gaps between different actions are not allowed.

To segment a meeting into a sequence of group meeting actions, we first extract features from the multimodal recordings, then construct statistical models that represent the meeting action sequence in terms of the extracted features. We have used four main categories of features outlined in section 3.4.1: prosodic features (such as fundamental frequency), speaker turn features, lexical features (based on a word-level transcription for each speaker), and motion-based video features. This feature extraction step may be regarded as describing a meeting as a set of streams, where each stream corresponds to a particular modality. To model this *multistream* situation, we have used dynamic Bayesian network (DBN) models in which a hierarchical state space is constructed, enabling individual feature streams to be processed independently at a lower, sub-action level, and collectively at a higher meeting action level. As previously outlined in chapter 2, Dynamic Bayesian Networks (and graphical models in general) present several advantages over Hidden Markov Models:

- increased flexibility in the state-space factorisation and structuring;
- increased capability to integrate some problem specific knowledge into the model, and therefore ability to develop potentially more discriminative models;
- improved and more parsimonious use of the parameter space;
- unified graphical-mathematical formalism.

Moreover it is possible to express simpler models such as HMMs and Kalman filters, or richer models including coupled HMMs, factorial HMMs, hierarchical HMMs, and semi-Markov models as DBNs (Smyth et al., 1997; Murphy, 2002a; Bilmes, 2003).

5.2 Single stream based meeting models

Several approaches to group action recognition have been proposed (section 4.3). A straightforward approach to the problem would consist of early integration of feature streams extracted from different subjects and modalities, followed by a simple HMM based infrastructure, and such an approach has formed the baseline system used in our experiments (section 5.4.3). This solution is simplistic since two main issues are disregarded: the explicit modeling of the interaction between multiple feature families, allowing an independent tuning and a better control over each feature stream; the necessity of relaxed temporal synchronisation constraints among multiple modalities and participants. Therefore coupled HMMs (Al-Hames and Rigoll, 2005b), layered HMMs (Zhang et al., 2006), and other multistream approaches are potentially better suited to this task. In particular, multistream models are highly flexible, intuitive and lend themselves to further improvement.

5.3 Multistream meeting models

Multistream approaches to group action recognition may use participant-based integration, or modality-based integration. In participant-based approaches, features from different modalities (individually extracted from each participant) are grouped together and modeled as a single stream. Thus each stream corresponds to a participant, and the whole group behaviour is inferred from the integration of single participant behaviours (sub-states). On the other hand, the modality-based approach focuses on modeling each communicative modality individually, grouping together behaviours associated with different participants.

Our multistream approach is based on processing different modalities independently. We assume that the group acts as a single subject and that “meeting actions” are related in the first instance to the entire group behaviour. Note that features such as “speaker turns” (section 3.3.2) are inherently related to the whole group rather than to individual participants. Moreover we preferred this strategy because it seems to provide better results when compared with the participant based one (McCowan et al., 2005).

A third, hybrid approach, obtained by modeling each participant-based uni-

modal feature stream independently, could be investigated also. Unfortunately, although this approach has promising results, it requires a much larger state-space (and hence considerable computing resources) for realistic applications.

5.3.1 Multistream DBN based model

The most attractive feature of the DBN framework is its extreme flexibility in the factorisation and structuring of the state-space (chapter 2). We assume that meeting actions can be interpreted as sequences of atomic units (*subactions*), much as sentences are subdivided into sequences of words. Thus we propose a model which is structured as a hierarchy of three layers: complete meeting actions at the top, subactions in the middle and the observed feature streams at the bottom. Thus low level features are mapped into atomic subactions, which are themselves the building blocks of complete meeting actions.

Each feature family represents a single modality (even if extracted from multiple media). If we assume that multiple modalities are independent at a subaction level and interact only at the highest level, then the feature streams are integrated (avoiding artificially introduced forms of stream weighting) at the top level during the global meeting action recognition. Thus this may be regarded as a multistream approach, since feature-streams are processed independently using their own subactions.

These subactions are obtained in an unsupervised way as the result of a training process. Each subaction is expected to represent a cluster of feature vectors which is associated with a particular meeting behaviour and is dominated by a common underlying dynamic. There is no clear and immediate interpretation of subactions, and supervised approaches to obtain subactions could be extremely difficult and expensive.

The state-space factorisation property may be exploited via both a hierarchical decomposition and a feature based subdivision. Consider a DBN, with a local BN at each time t. The resulting model, shown in figure 5.1 (A), appears as a tree shaped structure in which the observable features Y^F, $F = [1, N]$ are individually connected to their subaction variables S^F which are further connected to the action node A. The hidden variables S^F and A are each characterised by their own dynamics, in which each node is linked with its predecessor, forming a Markov chain.

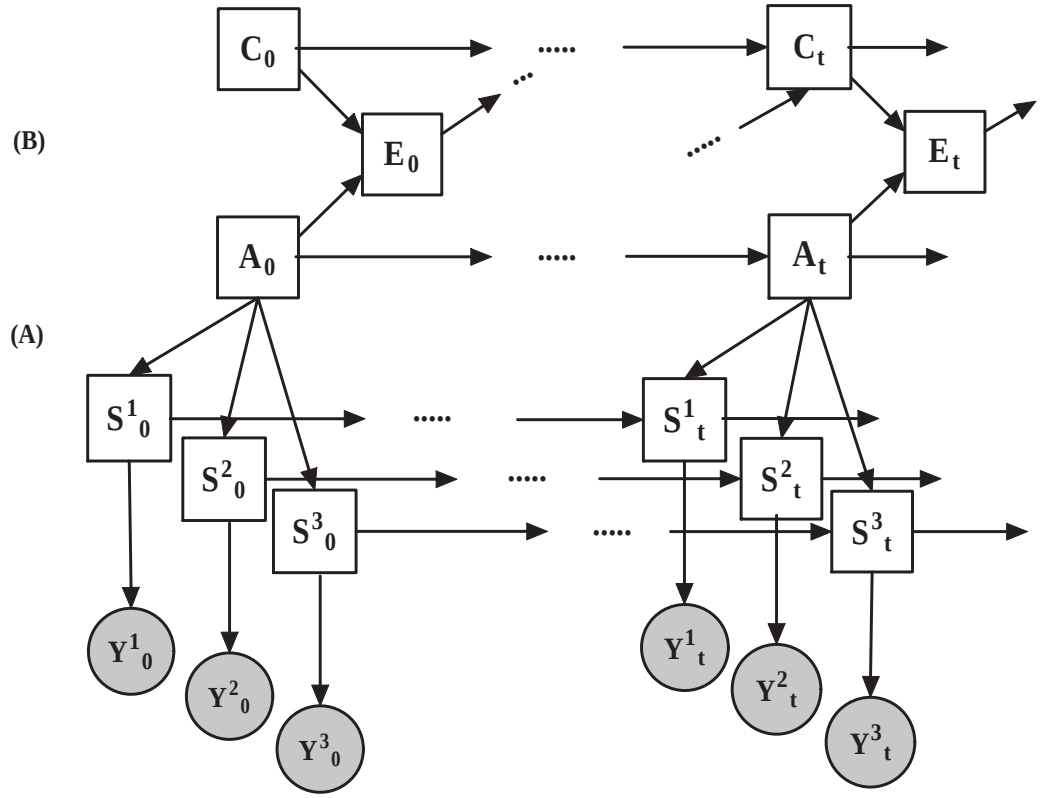


Figure 5.1: Multistream DBN model (A) enhanced with a "Counter Structure" (B). Square nodes represent discrete hidden variables and circles must be regarded as continuous observations.

The hierarchical relationship between A and S^F results in a structure that resembles the Hierarchical HMM (HHMM) topology (Fine et al., 1998) introduced in section 2.4.4. However this model is quite different, since HHMMs are characterised by a structured hierarchy of multiple Markov chains, and by a re-synchronisation mechanism which enables state transitions in higher chains only when lower HMMs have reached a "terminal state". Our model is free from this constraint, since actions A are free to change independently of the state of S^F. Similarly, the multistream approach to audio-video speech recognition (Dupont and Luettin, 2000) also relies on some re-synchronisation points, referred to as anchor points. It is possible to interpret this model as a Dynamical Systems Tree (Howard and Jebara, 2004) with three leaves, a single level of "aggregating nodes", and without the switching linear dynamical systems to couple the leaf nodes with subaction chains.

The lowest level of the model contains N continuous observable feature vectors

(nodes Y^F), each of which represents a single modality that has been extracted from raw audio/video recordings. Each feature stream Y^F is then mapped into discrete substates S^F through a Gaussian mixture model with M_F components:

$$P(Y_t^F = y \mid S_t^F = i) = \sum_{m=1}^{M_F} C(F,m,i)\mathcal{N}(y;\mu_{F,m,i},\Sigma_{F,m,i}) \tag{5.1}$$

where $\mathcal{N}(y;\mu_{F,m,i},\Sigma_{F,m,i})$ is a Gaussian density with mean $\mu_{F,m,i}$ and covariance $\Sigma_{F,m,i}$, evaluated at y, and $C(F,m,i)$ is the conditional prior weight of each mixture component m associated with stream F.

Each substate node S^F, $F = [1,N]$ is part of an independent Markov chain, and each subaction node S^F is a child of the global action node A. Therefore substate transition matrices $R_k^F(i,j)$ and an initial state distributions $\tilde{\pi}_k^F(j)$, associated with S^F, are functions of the action variable state $A_t = k$:

$$P(S_t^F = j \mid S_{t-1}^F = i, A_t = k) = R_k^F(i,j) \tag{5.2}$$

$$P(S_1^F = j \mid A_1 = k) = \tilde{\pi}_k^F(j) \tag{5.3}$$

$\tilde{\pi}_k^F(j)$ is the initial subaction distribution for the stream F, given an initial action $A_1 = k$; and $R_k^F(i,j)$ represents the transition probability from subaction i to substate j, given that the global meeting action variable ($A_t = k$) is in state k.

The sequence of action nodes A forms a Markov chain with multiple subaction nodes S^F as children. Therefore A can be regarded as a HMM generating N hidden discrete subaction sequences S^1, S^2, S^3, ... S^N through $R_k^1(i,j)$, $R_k^2(i,j)$, $R_k^3(i,j)$, ... $R_k^N(i,j)$ respectively. In a further analysis $R_k^N(i,j)$ is then responsible for the modelling of the joint dynamics of N multiple streams. $P(A_1 = i) = \pi(i)$ is the initial state probability vector associated with A, and $P(A_t = j \mid A_{t-1} = i) = Q(i,j)$ is the transition probability matrix. Note that the Markov chain A acts as an integration point, collating together the whole information carried by each subaction stream (representing a single feature family). Pushing the integration point to the highest level of the model in this way is referred to as "late integration".

Finally, the joint distribution for a sequence of T temporal slices, considering

the entire multistream model (figure 5.1 (A)), is given by:

$$P(A_{1:T}, S^1_{1:T}, ..., S^N_{1:T}, Y^1_{1:T}, ..., Y^N_{1:T}) = P(A_1) \cdot \prod_{F=1}^{N} \left\{ P(S^F_1 \mid A_1) \cdot P(Y^F_1 \mid S^F_1) \right\} \cdot \prod_{t=2}^{T} \{ P(A_t \mid A_{t-1}) \cdot \cdot \prod_{F=1}^{N} \left\{ P(S^F_t \mid S^F_{t-1}, A_t) \cdot P(Y^F_t \mid S^F_t) \right\} \} \quad (5.4)$$

Note that the cardinality of action nodes A is imposed by the size of the "action dictionary", $|A| = 8$ in this work, having: 4 types of monologue (M1, ..., M4), dialogue (DI), note-taking (NT), presentation (PR), and presentation at the whiteboard (WH). The cardinalities of the subaction nodes S^F are model parameters; from some development experiments we discovered that all our feature families (except the lexical based monologue/discussion discriminator) perform at their best when modeled with at least 5 subactions.

The number of free parameter of the multistream model depends on: the cardinalities $|A| = 8$ and $|S_F| = 5$, the number of Gaussian components M_F, and the size n_F of each feature vector F. The Conditional Probability Table associated to the action nodes A contains $|A|^2 = 64$ entries, and each subaction S^F CPT includes $|A| \cdot \left|S^F\right|^2 = 8 * 25 = 200$ elements. $|A|^2 + 3 \cdot |A| \cdot \left|S^F\right|^2 = 664$ parameters are required to encode the CPTs of the proposed 3-stream model, thus the overall model size is principally determined by the number of actions $|A|$. The number of free parameters needed to represent GMM means and variances can be estimated as $2 \cdot M_F \cdot n_F \cdot S_F$. Adopting two Gaussian components ($M_F = 2$) for each stream and having a total of $n_F = (6^3) + (12 + 1) + (2 * 2 * 4) = 245$ features, a total of $2 * 2 * 245 * 5 = 4900$ free parameters is required to encode means and variances. Since four Gaussian mixtures are assigned to each of the three streams, 12 additional parameters are needed to encode the conditional prior weights $C(F, m, i)$. Finally the proposed 3 stream model includes a grand total of $664 + 4900 + 12 = 5576$ free parameters.

5.3.2 Counter Structure

HMMs are characterised by a distribution in which the probability of remaining in a given state decreases as an inverse exponential (Rabiner, 1989). This state dura-

tion distribution is not well-matched to the behaviour of meeting action durations. This issue may be addressed in various ways, such as semi-Markov models (Ferguson, 1980; Russell and Moore, 1985; Murphy, 2002b), and state duplication to impose minimum duration constraints (Bourlard and Morgan, 1993; Lathoud and McCowan, 2003), as well as ad-hoc solutions such as action transition penalties.

We preferred to improve the flexibility of state duration modelling, by enhancing the existing model with an additional "counter structure" as in figure 5.1 (B). The duration of meeting actions is constrained by using a counter node C and an enabler node E. The sequence of counter nodes C forms a Markov chain, which attempts to model the expected number of recognized actions, whereby C is ideally incremented by a unit during each action transition. In this counter structure enhanced model, action variables A are not only parents of subactions S^F, but also of the enabler nodes E. Therefore A generates both N sequences of subactions S^F and a sequence of hidden enabler states E. Moreover the binary enabler variables E, reach their active state 1 only in the presence of action transitions ($E_t = 1$ only if $A_t \neq A_{t-1}$ and therefore $C_t = C_{t-1} + 1$), thus providing an interface between action variables A and counter nodes C. The counter variable C can be incremented only if the enabler variable E was high ($E_{t-1} = 1$) during the previous temporal slice $t-1$, as defined in the following deterministic relationship:

$$\begin{aligned} P(C_t = i+1 \mid C_{t-1} = i, E_{t-1} = 1) = 1 \\ P(C_t = i \mid C_{t-1} = i, E_{t-1} = 0) = 1 \end{aligned} \tag{5.5}$$

where $P(C_t = j \mid C_{t-1} = i, E_{t-1} = f)$ represents the state transition probability for the counter variable C given the global *counter structure* state during the previous frame $t-1$. Any evolution of the enabler node E is conditioned on both the action variable A and on the counter variable C. If A is in state k and the counter C in state j, the probability to activate E is given by:

$$P(E_t = f \mid C_t = j, A_t = k) = D_{j,k}(f) \tag{5.6}$$

where $D_{j,k}(f)$ represents the state transition probability associated with E. Suppose that the j^{th} meeting action has been recognised at time t ($A_t = k$), then the probability of encountering a new action (the $(j+1)^{th}$) or equivalently to have E activated ($E_t = 0$, $E_{t+1} = 1$) will be modelled by $D_{j,k}(f)$. Assuming that action transitions

are not possible during the first time frame $t = 0$, the initial probability of E is equal to $P(E_1 = 0) = 1$ and for coherence $P(C_1 = 0) = 1$.

The conditional probability tables (CPTs) associated to the "counter structure" require $2 \cdot |C| \cdot |C|$ free parameters for the counter variable C, and $2 \cdot |C| \cdot |A|$ for the enabler variable E. Assuming that each meeting includes a maximum of 10 meeting actions ($|C| = 10$), the 3 stream model outlined in the previous section requires 360 additional free parameters to encode the "counter structure". Note that the adoption of an enabler variable E within the "counter structure" has also the effect to reduce the dimension of the CPTs. Removing this variable (nodes E) and integrating (5.5) and (5.6) into a $P(C_t \mid C_{t-1}, A_{t-1})$, the number of parameters required by the "counter structure" will be increased by a factor:

$$\frac{|C|\,|A|}{2(|C| + |A|)} \quad . \tag{5.7}$$

The complete joint distribution of the multistream model enhanced with a counter structure (figures 5.1 (A) and (B) combined), computed for a sequence of T frames, is given by:

$$\begin{aligned} P(A_{1:T}, C_{1:T}, E_{1:T}, S^1_{1:T}, ..., S^N_{1:T}, Y^1_{1:T}, ..., Y^N_{1:T}) = \\ P(A_1) \cdot P(C_1) \cdot P(E_1) \cdot \prod_{F=1}^{N} \{P(S^F_1 \mid A_1) \cdot P(Y^F_1 \mid S^F_1)\} \cdot \\ \cdot \prod_{t=2}^{T} \{P(A_t \mid A_{t-1}) \cdot \; P(C_t \mid C_{t-1}, E_{t-1}) \cdot P(E_t \mid C_t, A_t) \cdot \\ \cdot \prod_{F=1}^{N} \{P(S^F_t \mid S^F_{t-1}, A_t) \cdot P(Y^F_t \mid S^F_t)\}\} \end{aligned} \tag{5.8}$$

Note that the use of a counter structure is not limited to the multistream model adopted here, but can be applied to any Markov chain.

5.4 Experimental results

In the previous sections three approaches for the automatic meeting structuring (baseline HMM, multistream-DBN, and counter enhanced multistream-DBN) have been proposed. Experimental results achieved using the proposed approaches are compared in sections 5.4.3 and 5.4.4, and a further extended multistream model is

outlined in section 5.4.5. All experiments were conducted on a subset of the publicly available M4 meeting data corpus described in section 3.2.1, using a dictionary of five group meeting actions resulting in eight distinct symbols: monologue (per speaker), dialogue, note-taking, presentation, and presentation at the whiteboard.

Only 30 meetings of the M4 corpus were transcribed (about 150 minutes), which is a too small amount of data to provide separate training and test sets. We therefore performed our experiments using a leave-one-out cross-validation strategy, in which models were trained on 29 meetings and tested on the remaining one; the procedure being iterated 30 times [1].

5.4.1 Feature setup

Our meeting action recognition experiments employed the four feature families presented in section 3.4.1: 12 prosodic, 216 speaker turn, 1 lexical and 16 visual features for total of 245 features. The monologue-dialogue discriminator (section 3.3.3.1) requires word level transcriptions. However this data comprises natural speech from non-native speakers, recorded using lapel and far field microphones, which results in high automatic speech recognition (ASR) word error rates. Our experiments were therefore performed using the reference orthographic transcriptions, and the reported results are for a semi-automatic system. ASR transcriptions of each speaker would be required for a fully automatic framework.

5.4.2 Testing conditions and performance evaluation

The task of meeting action recognition involves both segmentation and classification. Since the boundaries between meeting actions are not always precise, we have adopted an evaluation metric focused on the recognition of the correct sequence of actions and flexible about temporal boundaries (McCowan et al., 2003), the Action Error Rate (AER):

$$AER = \frac{Substitutions + Deletions + Insertions}{Correct\ number\ of\ actions} \cdot 100 \quad .$$

[1] Compared with the experimental setup in Dielmann and Renals (2004a) here we used a different sub-set of the M4 meeting corpus, a more robust experimental methodology (cross-validation), and a smaller parameter space ($\left|S^F\right| = 5$ instead of 7).

The AER is evaluated by summing the substitution, insertion and deletion errors of each recognised sequence when aligned to its reference transcription [2]. Note that the adopted meetings follow a predefined sequence of actions (section 3.2.1) which constitutes the ground truth for our experiments. The use of "scripted meetings" provides unambiguous annotations in terms of meeting action labels at the price of vaguely defined action boundaries. The AER metric, focusing on the labels sequence rather than their temporal boundaries, is ideally suited to this experimental setup. Lower AERs represent better recognition performances with a lower bound close to zero when the automatic system is able to perfectly recover the original scripting structure. AER is analogous to the word error rate metric used in speech recognition, and similarly to word error rate, is usually more severe than the frame based accuracy.

Action error rate and recognition accuracy represent the most widely used evaluation metrics on group meeting action recognition (McCowan et al., 2003; Zhang et al., 2006; Al-Hames and Rigoll, 2005a,b; Al-Hames et al., 2007b; Reiter et al., 2007). Several evaluation metrics have been proposed for the evaluation of similar tasks involving text segmentation. These include the P_k (Beeferman et al., 1999) and the *WindowDiff* (Pevzner and Hearst, 2002) metrics, both focused on text segmentation tasks such as topic detection (Hsueh et al., 2006) and story segmentation (Rosenberg and Hirschberg, 2006). The P_k metric estimates the probability that a randomly chosen pair of words, which are k words distant, is inconsistently classified with respect to the ground truth. *WindowDiff* is a variant of the P_k metric aimed at penalising false positives and near misses on an equal basis. Text segmentation boundaries can be placed only between words, thus both P_k and *WindowDiff* rely on the orthographic transcription. The meeting action recognition task is free from this assumption: meeting action boundaries are not necessarily related to the orthographic transcription [3]. Moreover P_k and *WindowDiff* aim to evaluate the segmentation quality ignoring the recognised meeting action labels.

This issue is directly addressed by the recognition metrics developed for joint dialogue act segmentation and classification, such as the NIST-SU, DER, strict, and

[2] Alignment, scoring and MAPSSWE significance testing performed using the NIST SCLITE Scoring Package, freely available from: `http://www.nist.gov/speech/tools/`.

[3] In all our meeting action recognition experiments we have adopted a frame-length of 0.5 seconds, thus a single observation can span over multiple words.

lenient metrics (section 7.7.1). However DA recognition is defined as a text segmentation problem, and words constitute the temporal units for the scoring process.

Meeting action recognition aims to obtain an abstract representation of the meeting structure, characterised by smoothed transitions between adjacent meeting actions, which may last a few seconds. Meeting action boundaries are vaguely defined and difficult to pinpoint in time. The employment of metrics such as accuracy and action error rate, derives from their ability to cope with imprecise temporal annotations, by focusing on the correct recognition of meeting action sequences.

5.4.3 HMM baseline results

A baseline system to relate low-level features with high-level meeting actions was developed using an ergodic HMM. Six systems were developed, one trained on each of the four feature sets invididually, one trained combining non visual features only, and a sixth using all four feature sets combined together. Since the four feature sets previously outlined were extracted in different contexts, they have different sampling rates. In order to share the same sampling frequency all of them were down-sampled, to a common sampling rate of 2 Hz. The word level based time-scale of lexical features was converted using the word time boundaries provided by transcriptions. Although the feature families shared the same sampling frequency after this process, it is not the case that they show similar temporal behaviours: each feature set has its own time-scale and level of asynchrony.

Tests on a development set (without the lexical information) indicated that an 11-state ergodic HMM was well-suited to this data. Table 5.1 shows the action error rates for each feature set. It can be seen that speaker turns provide the highest percentage of correctly recognised actions, followed by lexical features and prosodic features. Lexical features are most useful for discriminating between discussion and monologue, and the video-related features help most to discriminate between highly visual actions (note taking, presentation and presentation at the white-board). Note that monologue and discussions represent the 66% of the corpus, with the other actions comprising only 34%. All the results shown in table 5.1 are thus affected by this action distribution: speaker turn and lexical feature results are enhanced and video features weakened. The integration of visual features (last line of table 5.1) into the baseline system composed by speaker turn, lexical and prosodic features

Feature	Corr.	Sub.	Del.	Ins.	AER	(Var.)
Speaker turn features alone	65.4	16.7	17.9	20.5	55.1	(15.3)
Lexical features alone	58.3	23.7	17.9	7.1	48.7	(13.2)
Prosodic features alone	50.0	21.8	28.2	9.6	59.6	(19.4)
Turn, lex. and pros. features	**71.5**	**10.3**	**19.2**	**14.7**	**44.2**	**(18.3)**
Video features alone	48.1	21.8	30.1	7.1	59.0	(19.0)
All 4 feature families	**71.2**	**10.3**	**18.6**	**14.7**	**43.6**	**(16.1)**

Table 5.1: Comparison between meeting action recognition rate (% correct) and (substitution, insertion, deletion and overall) error rates achieved using four feature configurations and a simple HMM model. Action error rate variance (Var.) across folds was reported in brackets.

(fourth line of table 5.1) resulted in a small improvement in the overall recognition rate.

5.4.4 Multistream model

We compared experimentally the accuracy of the baseline HMM system with the multistream DBN model (section 5.3.1), and the multistream model enhanced with a counter structure (section 5.3.2). The multistream models were trained using three independent feature streams. Note that prosodic and lexical features were early integrated into a single 13 dimensional feature vector Y^3, and that the state-space has been limited to only five subactions per stream ($\left|S^1\right| = \left|S^2\right| = \left|S^3\right| = 5$). The multistream model shows a decisive improvement over this baseline system: the recognition rate (% correct) is increased by 17.9%, and together with a significant drop in the number of insertions, this results in a substantially reduced AER of 13.5%. Further small improvements were provided by the addition of a counter structure. This halved the number of insertions (at the cost of a small increase in the number of deletions), indicating an increased state duration, resulting in a further improvement in AER (12.2%). Both the multistream and the counter enhanced multistream model are significantly different from the baseline HMM system at the confidence level $p = 0.001$ according to the Matched Pair Sentence Segment Word

Model	Corr.	Sub.	Del.	Ins.	AER	(Var.)
HMM	71.2	10.3	18.6	14.7	43.6	(16.1)
multistream	89.1	3.2	7.7	2.6	13.5	(12.3)
multistream + counter	89.1	2.6	8.3	1.3	12.2	(12.8)

Table 5.2: Action error rates (%) and their across folds variances for: a simple HMM, a 3-streams DBN model, and a 3-streams counter enhanced version. Lower AERs indicate better performances.

Error (MAPSSWE) significance test (Pallett et al., 1990; Jurafsky and Martin, 2008) as implemented in the NIST Sclite Scoring Toolkit. This parametric test focuses on the difference between the number of errors that two system produce, averaged on a number of segments [4].

To further analyse the results, we give the confusion matrices for the multistream model enhanced with a counter structure (table 5.3). It is evident that the note taking action being the least frequent action (only 1.18% of the available corpus) is the most confused symbol. Monologues and presentations at the white-board are the better represented actions, and also the ability to discriminate between monologues and dialogues is excellent.

Model training is about three times slower than real-time on a 3GHz P4 processor, and feature decoding/recognition is two times faster than real-time. However, the memory requirements of Viterbi decoding were large, with about 1.5Gb required for decoding a system that used five sub-states per stream (section 5.3.1).

5.4.5 Extended multistream model

The binary lexical features are able to discriminate between monologue and discussion with a good accuracy (section 3.3.3.1). Since these two categories are a subset of the action dictionary, there is no reason why they need to be integrated with prosodic features and then modeled by an intermediate Markov chain (subaction S^F). Hence we have investigated an extended model (model $\mathcal{A}$ in figure 5.2) in which observable lexical features Y^4 are direct parents of the top level action chain

[4] A segment consists of a sequence of meeting actions.

	M1	**M2**	**M3**	**M4**	**DI**	**NT**	**PR**	**WH**	**INS**
M1	9								
M2		12							1
M3			12						
M4				10					
DI	1				40	1			
NT									1
PR						1	13		
WH							1	13	
DEL			1		5	4	2	1	

Table 5.3: Confusion matrix of recognised meeting actions for the counter enhanced multistream model, showing monologues (M1, ..., M4), dialogues (DI), note taking (NT), presentations (PR), presentations at the white-board (WH), insertion errors (INS) and deletion errors (DEL). Columns show desired symbols and rows obtained actions. Empty cells represent zero values.

(nodes A). The whole joint distribution after unrolling the model for T frames is given by a slightly modified version of equations (5.4) or (5.8). The number F of independent streams is set to $F = 3$, and $P(A_t \mid A_{t-1})$ is replaced by $P(A_t \mid A_{t-1}, Y^4)$. Note that speaker turns, prosodic features, and motion data are modeled as usual using three independent subaction Markov chains F with the following cardinalities: $\left|S^1\right| = 5$, $\left|S^2\right| = 5$, and $\left|S^3\right| = 5$. As can be seen in table 5.4 the AERs obtained using this model are poorer than the standard multistream approach discussed below, supporting the need of a dedicated intermediate level (subaction nodes S^F) for lexical feature processing.

In order to further address this issue, we investigated a second extended model (model $\mathcal{B}$ on the right side of figure 5.2) based on the multistream approach (figure 5.1). The lexical feature data stream was modeled in conjunction with prosodic data using a sub-state chain S^3, that was directly related to action nodes A. Similar to model $\mathcal{A}$ the joint probability distribution could be obtained from equation (5.8) by replacing $P(A_t \mid A_{t-1})$ with $P(A_t \mid A_{t-1}, Y^4)$. Note that Y^3 is the prosodic feature vector (as for the previous experiment) and Y^4 contains only the binary lexical fea-

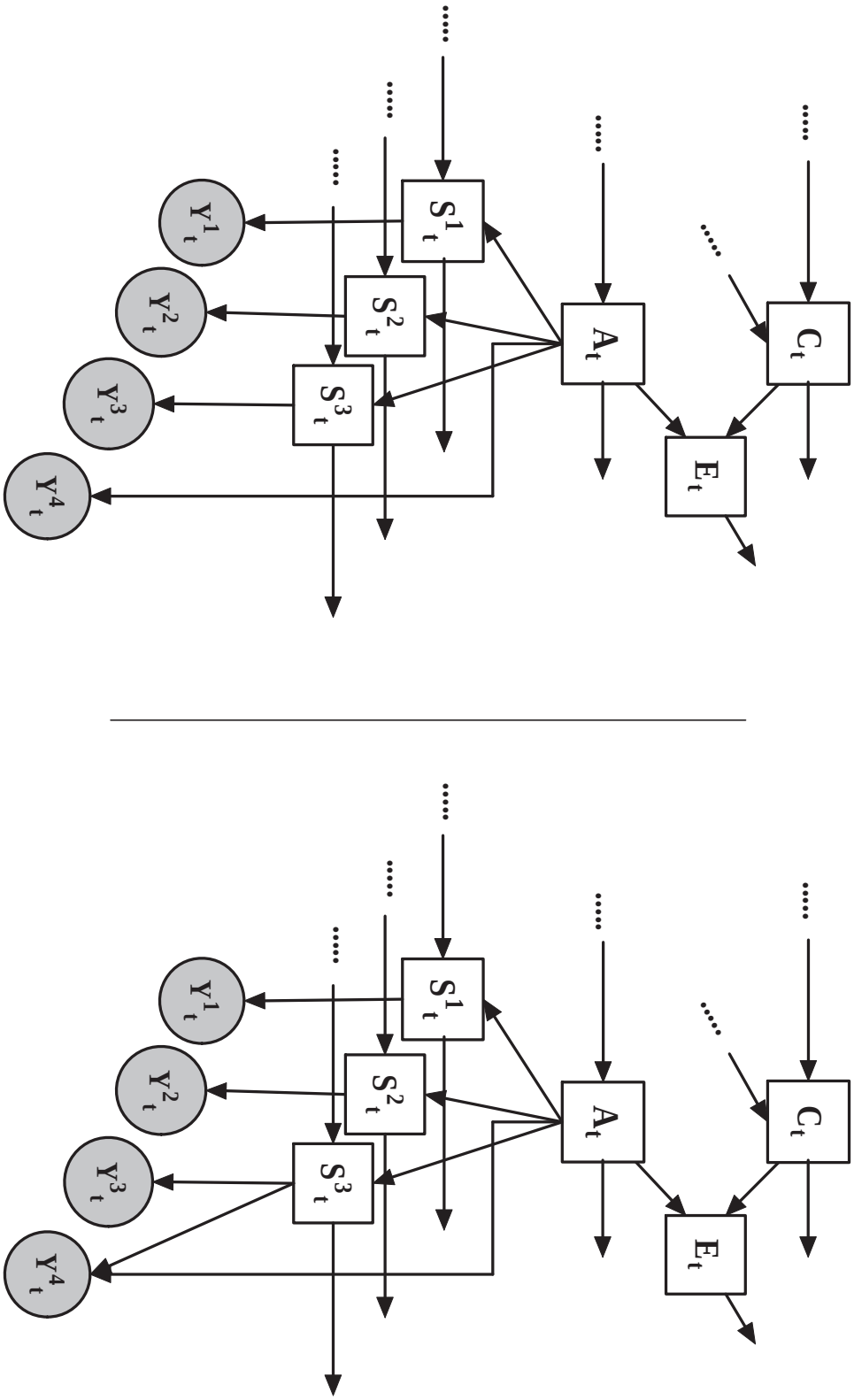


Figure 5.2: Extended multistream model $\mathcal{A}$ (left), and $\mathcal{B}$ (right). Both models are enhanced with a "Counter Structure".

Model	Corr.	Sub.	Del.	Ins.	AER	(Var.)
model $\mathcal{A}$	87.2	4.5	8.3	4.5	17.3	(14.9)
model $\mathcal{A}$ + counter	87.8	2.6	9.6	2.6	14.7	(14.0)
model $\mathcal{B}$	89.7	2.6	7.7	1.9	12.2	(12.3)
model $\mathcal{B}$ + counter	90.4	2.6	7.1	2.6	12.2	(11.0)

Table 5.4: Action error rates (%) and their variances across folds for two extended versions of the multistream model.

ture. The experimental results achieved with this model are reported in the last two rows of the table 5.4: the extended model $\mathcal{B}$ has a lower AER compared with $\mathcal{A}$, but the counter structure does not seems to improve the AER for model $\mathcal{B}$. MAPSSWE significance tests showed that all models ($\mathcal{A}$, $\mathcal{A}$+counter, $\mathcal{B}$, $\mathcal{B}$+counter) are significantly different from the baseline HMM system at $p = 0.001$. Model $\mathcal{A}$ and $\mathcal{B}$ are only significantly different at level $p = 0.05$. The standard multistream model and model $\mathcal{B}$ are significantly different at level $p = 0.01$.

Unfortunately the meeting corpus adopted for these experiments is limited, and it is not possible to discriminate between the standard multistream model and model $\mathcal{B}$. Comparing the confusion matrices of these two approaches (table 5.3 and 5.5 respectively) it is evident that the new model $\mathcal{B}$ tend to hypothesise more "dialogue" segments at the price of a few "monologue" deletions. These two models offer a similar accuracy, and the addition of a direct dependency of the highest level Markov chain on a low-level feature stream, did not compromise the overall performances.

In the following section 5.5 we will outline a feature comparison experiment conducted as part of a joint effort with the IDIAP and TUM research teams (Al-Hames et al., 2006a). The aim of this work was to present and compare different feature sets and approaches to the automatic meeting action recognition task.

5.5 Systems and features comparison

In a joint work (Al-Hames et al., 2006a) we presented a comparison of four approaches for the M4 meeting action recognition task, including the layered HMM

	M1	**M2**	**M3**	**M4**	**DI**	**NT**	**PR**	**WH**	**INS**
M1	9								
M2		11							
M3			12						
M4				10					
DI					42	2			2
NT									
PR						1	14		1
WH							1	13	1
DEL	1	1	1		3	3	1	1	

Table 5.5: Confusion matrix of recognized meeting actions for the extended multistream model $\mathcal{B}$ integrated with the counter structure.

(Zhang et al., 2006), our multistream DBN model (section 5.3), the higher level semantic approach (Reiter and Rigoll, 2004, 2005), and the mixed-state DBN for disturbed data (Al-Hames and Rigoll, 2005a,b). All these approaches provided comparable good performances.

The goal of this joint paper was also to compare three independently developed feature sets. Therefore in Al-Hames et al. (2006a) we have reported meeting action recognition experiments applying our multistream DBN model (section 5.3) to three multimodal feature sets: IDIAP, TUM, and UEDIN. The IDIAP set (Zhang et al., 2004a), kindly provided by the IDIAP Research Institute, Switzerland, comprises visual features (such as head and hand positions), prosodic features (F0, energy, and rate-of-speech), and sound source localisation related features (SRP-PHAT). The TUM feature set, generously provided by the Technische Universität München, includes global motion visual features and speaker diarisation related features (binary speech and silence segmentation). The UEDIN feature collection includes all the features presented in section 3.4.1 with the exception of visual and lexical features [5]. The complete list of features and the 3 different feature sets (IDIAP, TUM, UEDIN) are listed in table 5.6.

[5] These features are available only for 30 out of the 53 meetings used in this feature comparison experiment.

		Description	IDIAP	TUM	UEDIN
Person-Specific Features	Visual	head vertical centroid	✓		
		head eccentricity	✓		
		right hand horizontal centroid	✓		
		right hand angle	✓		
		right hand eccentricity	✓		
		head and hand motion	✓		
		global motion features from each seat		✓	
	Audio	SRP-PHAT from each seat	✓		
		speech relative pitch	✓		✓
		speech energy	✓		✓
		speech rate	✓		✓
		binary speech and silence segmentation		✓	
Group Features	Visual	mean difference from white-board	✓		
		mean difference from projector screen	✓		
		global motion features from whiteboard		✓	
		global motion features from projector screen		✓	
	Audio	SRP-PHAT from white-board	✓		
		SRP-PHAT from projector screen	✓		
		speaker turn features			✓
		binary speech from white-board		✓	
		binary speech from projector screen		✓	

Table 5.6: Audio, visual and semantic features, and the resulting three feature sets. Person-specific AV features have been extracted from individual participants, and group-level AV features are related to the whiteboard and projection screen regions.

Table 5.7 shows experimental results achieved using: a baseline ergodic 11-states HMM (section 5.2), a multi-stream approach with two feature streams, and the full counter enhanced multi-stream model (section 5.3). All the experiments depicted in table 5.7 were conducted on a subset of the M4 corpus (section 3.2.1) comprising 53 meetings [6] using a leave-one-out cross-validation procedure. The goal of these experiments was to compare different approaches and features on the largest amount of data, thus we decided to exclude from the comparison all the feature families which were not available for all the 53 meetings.

Our DBN based 2-stream approach (figure 5.1) has been tested in two different sub-action configurations: imposing $\left|S^1\right| = \left|S^2\right| = \{6\,or\,7\}$. Therefore four experimental setups were investigated; and each setup has been tested with 3 different feature sets, resulting in 12 independent experiments. The first feature configuration ("UEDIN") associates prosodic features and speaker activity features (sections 3.3.1 and 3.3.2) respectively to the stream S^1 and to S^2. The feature configuration labelled as "IDIAP" makes use of the multimodal features extracted at IDIAP, representing audio related features (prosodic data and speaker localisation) through the observable node Y^1 and video related measures through Y^2. The last setup ("TUM") relies on two feature families extracted at the Technische Universität München: binary speech profiles derived from IDIAP speaker locations and video related global motion features; each of those was assigned to an independent sub-action node. Note that in the HMM based experiment the unique observable feature stream Y has been obtained by merging together both the feature vectors Y^1 and Y^2 ("early integration").

Considering only the results (of table 5.7) obtained using the UEDIN feature setup, it is clear that the simple HMM shows much higher error than any other multi-stream configuration. The adoption of a multistream based approach reduces the AER to less than 20%, providing the lowest AER (11%) when sub-action cardinalities are fixed to 7. UEDIN features seem to provide a slightly higher accuracy if compared with IDIAP and TUM setups, but it is essential to remember that our DBN models have been optimised for the UEDIN features. Moreover overall performances achieved with the multistream approach are very similar (AER are

[6] Note that differences in the data-set, feature set, and state-space factorisation make a direct comparison between tables 5.2, 5.4, and 5.7 infeasible.

Model	Feature	Corr.	Sub.	Del.	Ins.	AER	(Var.)
HMM	UEDIN	63.3	13.2	23.5	11.7	48.4	(15.9)
	IDIAP	62.6	19.9	17.4	24.2	61.6	(15.4)
	TUM	60.9	25.6	13.5	53.7	92.9	(20.0)
2 streams $\left(\left\|S^F\right\|=6\right)$	UEDIN	86.1	5.7	8.2	3.2	17.1	(12.5)
	IDIAP	77.9	8.9	13.2	4.6	26.7	(17.1)
	TUM	85.4	9.3	5.3	6.8	21.4	(16.0)
2 streams $\left(\left\|S^F\right\|=6\right)$ + cnt	UEDIN	85.8	7.5	6.8	4.6	18.9	(13.2)
	IDIAP	79.4	10.0	10.7	4.3	24.9	(16.8)
	TUM	85.1	5.7	9.3	6.4	21.4	(14.0)
2 streams $\left(\left\|S^F\right\|=7\right)$	UEDIN	90.7	2.8	6.4	1.8	11.0	(11.6)
	IDIAP	86.5	7.8	5.7	3.2	16.7	(14.7)
	TUM	82.9	7.1	10.0	4.3	21.4	(17.0)

Table 5.7: Action error rates (%) and their variance across folds (Var.) for a HMM, and for a multi-stream (2 streams) approach with and without the "counter structure". The models have been individually tested on 3 different feature sets (UEDIN, IDIAP, TUM).

always in the range between 26.7% and 11.0%), and all of them may be considered promising. The TUM setup seems to be the configuration for which switching from a HMM to a multistream DBN approach provides the greatest improvement in performance: the error rate decreases from 92.9% to 21.4%. If with the UEDIN feature set the adoption of a counter structure is not particularly effective, with IDIAP features the counter structure provides a significant AER reduction (from 26.7% to 24.9%). Independently of the feature configuration, the best overall results are achieved with the multistream approach and a state space of 7 by 7 substates. All DBN approaches provide significantly different recognition outputs (MAPSSWE significance test at level $p = 0.001$) when compared to their corresponding HMM baselines. However no significant differences according to the MAPSSWE test were found comparing the three TUM multistream systems.

In the following section we will outline the latest advancements on meeting action recognition made by the research community.

5.6 Related work

Further progresses on the M4 group action recognition task have been recently published by Reiter et al. (2007). This work successfully applies a Hidden Conditional Random Field (HCRF) model (Quattoni et al., 2005) to the meeting structuring problem. HCRFs, similarly to conventional CRFs (section 7.8), avoid the assumption of conditionally independent observations typical of generative models. Moreover HCRFs further generalise CRFs by incorporating hidden variables and allowing them to deal with time-sequences. Note that conventional CRFs, as originally formulated in Lafferty et al. (2001), do not provide a way to estimate the conditional probability for an entire time-series (given a class label). The HCRF based meeting action recogniser outperformed a HMM on all the configurations attaining a recognition accuracy of 92.1%. This system achieved one of the best recognition results on the M4 corpus, defining the state of the art for this task.

Following a slightly different research direction, Al-Hames and Rigoll (2005a,b) addressed the automatic classification of meeting actions in presence of disturbed data. The M4 meeting corpus was artificially degraded: audio recordings were corrupted adding babble noise with 10dB SNR, and randomly placed gray bars covering one third of the picture were used to perturb the video streams. Three feature families, related to video, audio, and microphone array processing, were extracted on six relevant spatial positions (in analogy to section 3.3.2 and 3.3.4). The resulting feature set extends the TUM feature selection of table 5.6. Seven global motion features (motion center, dynamics, mean absolute deviation, and intensity) were extracted from the video streams. Four Mel Frequency Cepstral coefficients (MFCC) plus energy were extracted from the lapel microphone recordings, and 6 location based binary silence/speech features were estimated using the microphone array (section 3.3.2.1). Audio and visual feature streams were jointly modelled using a multistream approach based on a two chains coupled HMM [7], where the audio Markov chain generates acoustic observations through a conventional Gaussian Mixture Model, and the video stream is obtained through a Linear Dynamical System. Note that a LDS is a Kalman filter which uses the coupled audio-visual hidden state to smooth the video feature stream. Visual disturbances are reduced

[7] A partially coupled HMM based on three Markov chains: video, audio, and microphone array; has been adopted in Al-Hames and Rigoll (2005a).

using the Kalman internal state and further compensated by exploiting the audio information (through the audio Markov chain of the coupled HMM). Baseline HMM experiments, both on clean and corrupted data, highlighted the importance of sound source localisation, followed by acoustic related features and visual information. The HMM-LDS approach (mixed-state DBN model) outperformed the "early integration" HMM system on all testing conditions, both on clean and corrupted data. Further improvements using an Asynchronous Hidden Markov Model (AHMM) can be found in Al-Hames et al. (2007b). The AHMM (Bengio, 2003) models the joint probability of two observation streams even if the two sequences are not synchronised, have different lengths, or different sampling rates. Assuming two observation sequences Y_1 and Y_2, each hidden state x can emit just a symbol from Y_1, as in a conventional HMM, or jointly emit two symbols at the same time, one from Y_1 and one from Y_2. The adoption of an asynchronous model resulted in significant performance improvements over the mixed-state DBN model both on clean and disturbed video data.

Group meeting actions provide a clearly structured view of the conversation, being thus useful to archive and index meeting collections, send an automatically edited live video-stream to remote meeting participants, and control active sensors such as pan-tilt cameras. An investigation of different feature families for the automatic editing of meeting video footage has been presented in Al-Hames et al. (2006b). The goal is to select the most relevant "video mode" [8] in according to the evolution of the conversation. Note that naïve solutions such as showing only the currently active speaker result in frequent scene changes during a dialogue, and in concealing important visual feedback from the audience (e.g.: nodding in disapproval) during a presentation. Group actions represent a valuable cue for this task and provide a guideline to evaluate the automatic selection made by a "virtual meeting director". However group meeting actions lead to slow changing static footages, focused for example on the speaker giving a monologue. Conversely low level audio and video features result in unwatchable videos constantly switching between different video modes. In this case a smoother output can be obtained integrating the fast changing low-level features over a sliding temporal window.

The work outlined in Al-Hames et al. (2006b) focuses on individual feature

[8] Video modes correspond to single camera views or compositions of multiple views.

streams although a simple feature fusion experiment is also reported. A richer feature combination scheme based on a layered HMM is introduced by Al-Hames et al. (2007a). The lowest HMM layer recognises 14 individual actions (such as stand-up, sit-down, and nodding) using an audio-visual feature set composed by: global motion features, head and hand positions, and acoustic features (12 MFCC + energy + Δ + $\Delta\Delta$). The recognised individual actions, together with group related features (e.g.: motion in front of the whiteboard), are then exploited by the second HMM layer to learn the mapping between individual/group behaviours and desired video modes. This supervised approach relies on manually annotated examples: annotators were asked to select the sequence of camera views (video modes) that they thought would better represent the underlying meeting [9]. The same annotation can also be used to score the system output, comparing it to the reference camera view selection. On a joint segmentation and classification task based on 7 video modes, the proposed layered HMM framework clearly outperformed an "early integration" HMM system.

The automatic analysis of group interaction is not limited to the M4 group meeting action recognition task. For example Dai et al. (2007) proposed a multilevel probabilistic model for context aware computing within a meeting room. A event-driven multilevel DBN is employed to detect the sequence of group interactions such as presentation, discussion, and break. Note that each interaction class is hierarchically defined with multiple layers of sub-interactions (e.g.: a "presentation" is composed by "lecturing" and/or "question and answers" segments). The aim is to develop an online approach for the group interaction analysis, being able to provide attentive services during meetings such as controlling pan-tilt cameras. Although multimodal and dominated by the speech modality, a conversation can also be investigated limiting the attention to the visual content. Otsuka et al. (2006) proposed to analyse multiparty conversations using only the video recordings of a meeting. Head orientations are estimated from video sequences and adopted as gaze direction approximates. A Markov switching model is then used to structure the conversation in terms of gaze patterns: gaze convergence, dyadic-link, and divergence. This hierarchical DBN model (section 2.4.4) is composed by two layers which generate

[9] Not surprisingly this is a very subjective annotation task and the inter-annotator agreement is quite low.

two independent feature streams.

Automatic meeting action recognition is a fertile research area offering some further space for improvement. For example the group meeting action recognition task can be generalised to new meeting corpora, such as the AMI corpus, and extended by defining richer annotation schemes. The research on this subject lead to the concept of an “automatic video director”. This idea can be further extended by generating audio-visual summaries of a meeting, where the most salient excerpts are automatically selected and smoothly edited.

5.7 Discussion

In this chapter we have addressed the problem of automatically segmenting a meeting into a sequence of group meeting actions taken from a dictionary of events such as monologue, discussion, and presentation. We performed our experiments using a publicly available corpus of meetings recorded using multiple cameras and microphones. This corpus has some limitations (section 3.2.5), including the short duration of each meeting (5 minutes per meeting, on average), the fact that only 30 meetings (150 minutes) were fully annotated, and the somewhat artificial content of the meeting agenda and topics. Despite these limitations, the M4 corpus does feature natural and spontaneous interactions between participants, and provides a good basis for investigations in multimodal processing and event recognition in multiparty meetings.

The multi-perspective audio/video recordings were processed by extracting relevant multi-modal features, followed by statistical modeling. Four feature families were extracted from these recordings, representing speaker turn dynamics, prosodic and lexical information, and participant motion (head/hand/body movements). In order to relate these low-level feature streams with high-level meeting actions, a DBN multistream model was adopted. Using this multistream framework, it is possible to process each feature stream independently at a lower level of the model, and to collect together partial results at the upper stage of the model, thus offering a hierarchical approach to the integration of multiple feature streams.

The capability to incorporate some knowledge of the problem into the model structure is one of the principal features of the DBN framework, resulting in a more

parsimonious model compared with simple HMMs. Moreover the use of a multi-stream approach shows some advantages over merging all the feature families into a single feature vector (early integration):

- The integration point in which knowledge from different feature streams is collected together, may be delayed to a later stage of the processing (*late integration*).
- The independent feature processing increases the flexibility in modeling the interdependences between different modalities, allowing the model to encompass complex statistical dependences, lack of synchronism, and multiple time scales.

These advantages have resulted in a significant increase in accuracy when the DBN multistream models are used in place of a HMM for the meeting action recognition task, resulting in an action error rate of 12.2%.

The multistream DBN based approach for group meeting action recognition was also validated on three multimodal feature setups: a subset of the feature collection presented in section 3.4.1 (UEDIN), and two independent feature sets kindly provided by IDIAP and TUM research institutes. The proposed DBN model achieved good recognition accuracies on all the 3 feature setups, confirming the validity of this approach and proving its flexibility toward different feature sets.

Meeting actions aim at highlighting complex interactions between different meeting participants thus showing overall group intentions, for example: dialogues involve two or more meeting participants, a presentation given by a single participant is usually supported through the active feedback from the rest of the group, etc. In the following two chapters we will investigate the same communicative process by modelling the discourse structure at a fine grained level. Instead of detecting whole group behaviours we will concentrate on individual participant intentions, by segmenting the conversation in terms of “Dialogue Acts”.

Chapter 6

Dialogue Act recognition

6.1 Introduction

Dialogue acts (DAs) form a useful level of representation for the interpretation of conversations. A DA is a construct that describes the role played by an utterance in the conversation, and provides a bridge between the orthographic (word-level) transcriptions and a richer representation of the discourse. A conversation may be segmented into a sequence of DAs, with each DA assigned a label describing the function played by that utterance within the conversation. DA labels may incorporate syntactic, semantic and pragmatic factors. In addition to providing information about the structure of a dialogue and the course of a conversation, DAs are also able to capture, at a coarse level, individual speaker attitudes and intentions, their interaction role and their level of involvement.

Multiparty meetings were intensively researched over the past several years, with a growing focus on how a meeting may be automatically analysed and interpreted in terms of the group discourse and interaction. Example applications included automatic summarisation (Murray et al., 2006), topic segmentation and labelling (Galley et al., 2003; Hsueh and Moore, 2006), group action detection (Al-Hames et al., 2006a; McCowan et al., 2005; Dielmann and Renals, 2007a), participant influence (Rienks et al., 2006), and dialog structure annotation (Purver et al., 2007). The reliable recognition of the DA sequence in a meeting, and the resulting knowledge of the discourse structure, plays an important role in the development of such applications.

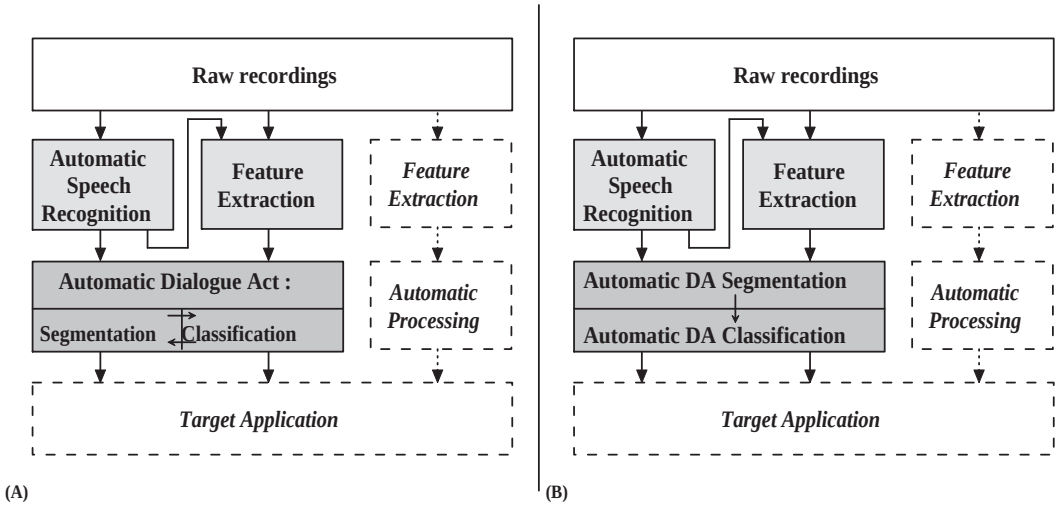


Figure 6.1: Automatic Dialogue Act recognition infrastructure, using a joint approach (A) or a sequential system (B). Word level transcriptions and prosodic features are automatically extracted from the raw audio recordings and then supplied to the DA recogniser.

6.2 Previous work on automatic Dialogue Act recognition

The dialogue act recognition task comprises two related sub-tasks: segmentation, and classification or tagging. These tasks may be performed jointly or sequentially (figure 6.1). In a sequential approach the conversation is first segmented into unlabelled DA segments, then each detected segment is tagged with a DA label. The joint approach:

- performs both tasks concurrently, detecting DA segment boundaries and assigning labels in a single step;
- is able to examine multiple segmentation and classification hypotheses in parallel, whereas only the most likely segmentation is supplied to the DA classifier in a sequential approach;
- is potentially capable of greater accuracy, since it is able to explore a wider search space, but the optimisation problem can be more challenging.

In a sequential system the two sub-tasks can be optimised independently. Note that an integrated system may be used as a segmenter by ignoring its classifications. For purposes of comparison, often it may also be used as a classifier, by forcing a human DA segmentation onto it. Most previous work concerned with DA modelling has focused on tagging presegmented DAs, rather than the overall recognition task which includes segmentation and tagging. Indeed, automatic linguistic segmentation (Stolcke and Shriberg, 1996; Shriberg et al., 2000; Baron et al., 2002) is often regarded as a research problem itself.

6.2.1 Automatic Dialogue Act tagging

The use of a generative HMM discourse model (Nagata and Morimoto, 1993), in which observable feature streams are generated by hidden state DA sequences, has underpinned most approaches to DA modelling, and a good overview of this approach is given by Stolcke et al. (2000). The discourse history is typically modelled using an n-gram over DAs, although approaches such as polygrams (Warnke et al., 1997) were tested. Lexical features were widely used for DA tagging (section 3.3.3), via cue words or statistical language models, including approaches such as multiple parallel n-grams (Venkataraman et al., 2005), hidden event language models (Zimmermann et al., 2006a), and factored language models (Ji and Bilmes, 2005). Several authors have previously investigated the use of prosody to disambiguate between different DAs with a similar lexical realisation (Bhagat et al., 2003), and investigated approaches to automatically select the most informative features (Shriberg et al., 1998; Hastie et al., 2002). Prosodic features such as duration, pitch, energy, rate of speech and pauses were successfully integrated into the processing framework (table 6.1).

Ji and Bilmes (2005) proposed a switching-DBN based implementation of the HMM approach above outlined, applying it to the DA tagging of ICSI meetings. They also investigated a conditional model, in which words generate dialog act labels (instead of having a generative framework where sequence of words are generated by the enclosing DA labels). DA tagging experiments were performed both using multiple parallel n-grams or adopting a FLM with two factors: word identities and DA labels. The generative approach prevailed over the conditional model, reporting the best classification accuracy when used in conjunction with a FLM. Since

this work used only lexical features, and a large number of DA categories (62), a direct comparison with the results reported by Ang et al. (2005); Zimmermann et al. (2006a,b); Dielmann and Renals (2007c) is not possible.

Venkataraman et al. (2003) proposed an approach to bootstrap a HMM-based dialogue act tagger from a small amount of labeled data, followed by an iterative retraining on unlabeled data. The DA tagger initially trained on a small amount of annotated data is then adapted and retrained on a much larger unannotated dataset. The proposed tagger makes use of the standard HMM framework, together with dialogue act specific language models (3-grams) and a decision tree based prosodic model. The authors also advanced the idea of a completely unsupervised DA tagger in which DA classes are directly inferred from data.

More recently, there have been a number of conditional models applied to DA classification including support vector machines (SVMs) (Fernandez and Picard, 2002; Liu, 2006) and maximum entropy classifiers (Venkataraman et al., 2005; Ang et al., 2005). Features for these models include both lexical and prosodic cues, as well as contextual DA information (Venkataraman et al., 2005) (table 6.1).

A framework for the automatic DA classification of the Spanish CallHome spontaneous speech corpus (using 8 DA labels) was outlined by Fernandez and Picard (2002). The proposed approach relies on a SVM based classifier and a set of features derived from energy and pitch contours. Numerical results demonstrated the importance of prosodic cues, highlighting that even in absence of an orthographic transcription it is still possible to detect DAs well above chance.

A maximum entropy based DA classifier for the 5 DA ICSI task (section 7.7.3) was proposed by Ang et al. (2005). DA length, first and last two word identities, and contextual features (initial word of the following DA unit) were included during the classification process, together with an extensive set of prosody related features. Since the MaxEnt approach requires binary features a decision tree (DT) was constructed for the continuous prosodic features. DA class posterior probabilities estimated with the DT were binned, reduced to binary values, and then provided to the discriminative classifier. The resulting classifier defines the state-of-the-art on the 5 DA ICSI tagging tasks, reporting a classification error rate (CER) of about 18.8% on reference transcriptions and of about 26.0% on ASR output [1].

[1] A direct comparison to this ASR condition will require the exact ICSI ASR automatic transcrip-

Liu (2006) proposed an automatic DA classifier based on the combination of multiple binary SVM classifiers via error correction output codes (ECOCs). This work extends the 5 DA ICSI tagging task outlined in Ang et al. (2005) comparing the originally adopted maximum entropy classifier with a multiclass SVM and 4 different setups based on ECOC SVM classifiers. All ECOC classifiers performed better than a multiclass SVM, but were outperformed by the baseline MaxEnt system of Ang et al. (2005).

Generative and conditional approaches can also be combined together. For example Surendran and Levow (2006) integrated local discriminative SVM classifiers (using prosodic and lexical features) within a HMM discourse model by applying Viterbi decoding to class posterior probabilities estimated using the SVMs. The SVM-HMM tagging system was applied to the 13 DA classes of the HCRC Maptask corpus (Carletta et al., 1997). This is a corpus of spontaneous task oriented conversations, consists of 128 dialogues between two participants interacting on a game-move task: a *giver* provide instructions to guide a *follower* through the route on a map. The approximately 15 hours of conversational speech were annotated in terms of DAs using a dictionary composed by 13 classes: instruct, explain, check, align, yes/no question, wh- question, acknowledge, reply yes, reply no, reply to a wh- question, clarify, ready and unlabelled.

In order to compare DA classification performances on different meeting data (Switchboard, ICSI and AMI) a portable DA tagger was developed by Verbree et al. (2006). The proposed system makes use of several feature families: manually annotated question marks, lexical cues, DA unit durations, compressed ngrams of both words and Part Of Speech tags; and a bigram discourse model. The extracted features are then modelled using the J48 classifier of the Weka toolkit (Witten and Frank, 2005). While the classification accuracy achieved on the Switchboard 42 DA task is about 5% lower than the state of the art, the system outperforms all the previous works on the 5 DA ICSI task, reaching an accuracy of 89.3% [2]. The classification accuracy on the AMI 15 DA is about 59.8% using reference orthographic transcriptions and 49.3% using the ASR output.

DA tagging experiments on the 15 DA AMI task (section 7.7.4), using a max-

tion, which is not publicly available.

[2]Manually annotated question marks were employed by this system, thus any direct comparison to previous works should be carefully considered.

imum entropy based classifier, were reported in Lesch (2005) and Lesch et al. (2005b). The proposed system adopts a wide set of features belonging to the following 5 classes: lexical features, DA unit length and duration, temporal relation between adjacent utterances, speaker change and dialogue act history. A feature selection algorithm, growing a feature subset by iteratively ranking candidate features according to their classification accuracy, was adopted to build a compact feature vector composed only by the most informative features. The best classification accuracy obtained on the AMI evaluation set is 65.8% for reference transcriptions and 54.9% with automatically recognised words (classifying ASR deleted DA units by chance). This result defines the state of the art for the 15 DA AMI tagging task. Similarly to Verbree et al. (2006) and Dielmann and Renals (2007b), when the reference orthographic transcription is replaced by an imperfect automatic transcription the classification accuracy falls by about 10% (absolute).

6.2.2 Features for automatic Dialogue Act processing

Table 6.1 outlines the most common features used in previous DA tagging and recognition studies. Four main feature classes can be defined:

Lexical features A wide range of features spacing from word identities and lexical cues (presence/absence of specific keywords) to specific lexical or grammatical patterns, or elaborate language models: multiple DA specific ngrams, Factored Language Models, polygrams, etc. Since fully automatic systems need to operate on ASR transcriptions, the impact of automatic transcription errors on the extracted lexical features should be carefully considered. DA unit length, intended as the number of words contained by the current DA segment, is a popular lexical related feature able to highlight typically short DAs, such as backchannels, or to support elaborate durational models (section 7.5.3).

Context features These features summarise the relationship between current and surrounding DA units, including cues such as word identities from the previous/next unit, and speaker turn related features.

Prosodic features A wide selection of acoustic related features extracted from the

Features	Ang et al. (2005)	Fernandez and Picard (2002)	Venkataraman et al. (2003)	Venkataraman et al. (2005)	Jurafsky et al. (1998)	Zimmermann et al. (2006a)	Zimmermann et al. (2005)	Warnke et al. (1997)	Ji and Bilmes (2005)	Surendran and Levow (2006)	Liu (2006)	Dielmann and Renals (2007b)	Verbree et al. (2006)
Sentence length/duration	✓			✓							✓	✓	✓
Annotated question marks													✓
First two word identities	✓			✓							✓		
Last two word identities	✓			✓							✓		
Bigram of the first two words											✓		
Specific cue words or phrases					✓								✓
Grammar patterns					✓								
Sparse bag of ngrams										✓			
Polygrams of words								✓					
Factored Language Models									✓			✓	
Part Of Speech ngrams													✓
Ngrams of words			✓	✓		✓	✓		✓		✓		✓
First word of the next segment	✓			✓							✓		
Speaker (turn) change			✓	✓						✓	✓		
Pitch		✓	✓		✓			✓		✓		✓	
Energy		✓			✓					✓		✓	
Duration		✓	✓		✓			✓		✓		✓	
Pauses			✓		✓			✓				✓	
Rate of speech										✓			
Ngrams of previous DA labels			✓	✓			✓		✓	✓		✓	✓

Table 6.1: Features used for DA segmentation and DA classification in different studies.

audio recordings (section 3.3.1): pitch contour and pitch slopes, signal energy, rate of speech, word and pause durations. These features had proven to be useful in disambiguating between multiple DA classes with a similar lexical realisation (Bhagat et al., 2003), and in facilitating the DA segmentation (section 7.7.4). An extensive study on the use of prosody for automatic DA classification can be found in Shriberg et al. (1998).

A discourse model Concentrate on the DA labels of the surrounding segments (section 7.4); usually only the preceding ones, which were already recognised, are taken into account. N-gram language models, or more elaborated models, are trained on the sequence of manually annotated DA labels from the training dataset. During the testing phase the discourse language model is then used to estimate the prior probability of a new DA label given the recognition history.

6.2.3 Automatic Dialogue Act recognition

An early system for the integrated joint DA segmentation and classification was outlined by Warnke et al. (1997). 18 DA classes are automatically recognised in short task oriented dyadic conversations (appointment scheduling of the German VERBMOBIL corpus). The system, using a multi-layer perceptron and a Language Model for segmentation, a polygram LM for DA classification, and a joint search algorithm to score multiple joint recognition hypotheses, reported a significant improvement over a sequential approach.

Ang et al. (2005) addressed the automatic dialogue act recognition problem using a sequential approach, in which DA segmentation was followed by classification of the candidate segments. Promising results were achieved by integrating a boundary detector based on *vocal pauses* with a hidden-event language model (HE-LM), a language model including dialogue act boundaries as pseudo-words. The dialogue act classification task was carried out using a maximum entropy classifier, together with a relevant set of textual and prosodic features. This system segmented and tagged DAs in the ICSI meeting corpus (using the 5 broad DA categories outlined in section 3.2.2), with good levels of recognition accuracy: 19.6% using the Lenient

metric, 64.4% with the Strict metric, and about 54.4% of DA Error Rate [3]. However results comparing manual with automatic transcriptions indicated that the ASR error rate resulted in a substantial reduction in accuracy (absolute 5% on the Lenient metric).

In a later work Zimmermann et al. (2006a) compared two joint approaches on the same experimental setup. An extended HE-LM able to predict not only DA boundaries but also the type of the DA, and a HMM recogniser inspired by HMM based part of speech taggers, were trained on lexical features and compared using several of the metrics discussed in section 7.7.1. The joint HE-LM system obtained lower recognition error rates than the HMM based DA recogniser, achieving performances closer to the discriminative sequential approach of Ang et al. (2005).

In Zimmermann et al. (2006b) the authors further extended the joint HE-LM DA recogniser. A discriminative maximum entropy DA boundary detector and tagger is trained on discretised inter-word pauses with a lexical context of 4 words. The weighed combination of classification probabilities for both systems (HE-LM and MaxEnt) provides the most likely sequence of labelled DA units, which is able to outperform the baseline sequential approach of Ang et al. (2005). The resulting system achieved a DA Error Rate of 51.0% and a Strict recognition error rate of 62.8%. Note that multiple concurrent DA segmentation and classification hypotheses could be evaluated by joint DA recognisers, enabling the investigation of larger search spaces compared with two-step sequential segmentation-classification approaches.

6.3 Applications of automatic Dialogue Act processing

Valuable insights into the discourse structure can be gained through the reliable recognition of the DA sequence in a conversation. This knowledge can be beneficial for the development of applications in a multitude of domains, including spoken dialogue systems, machine translation, automatic speech recognition, automatic summarisation, topic segmentation and labelling, action items detection, group action detection, participant influence detection, and dialogue structure annotation.

[3] DA recognition metrics are discussed in section 7.7.1.

As outlined in chapter 3, during the last decade, several corpora were annotated in terms of DAs, and a relevant literature on automatic DA recognition was developed (section 6.2). Several works also focused on the exploitation of automatically extracted DAs. Moving from the idea that the knowledge about the ongoing conversation (conveyed by DAs) can be used to enhance language modelling, improving Automatic Speech Recognition of conversational speech was one of the first targets. Jurafsky et al. (1997a) investigated the use of automatically detected DAs to improve automatic speech recognition. The 1155 pre-segmented conversations from the Switchboard database were automatically tagged using a clustered dictionary of 42 DA labels. The system made use of a generative DA tagging infrastructure based on: prosodic features (pitch, speaking rate, energy, etc.), word sequence based trigram models, and a bigram discourse language model. Automatic transcriptions were generated through ASR and then provided to the automatic DA tagger. The automatically detected DA classes were then used to rescore the ASR output by means of a novel *DA conditioned mixture Language Model*: n-best lists associated to test-set utterances were rescored using a mixture of DA specific LMs. Numerical results on the Switchboard corpus showed only a limited improvement (0.3%) on the ASR word error rate, principally because of the skewed distribution of DA classes (statements accounted for 83% of the corpus). However DA rescored ASR should have a larger space for improvement on specific tasks with more even DA distributions (i.e: task oriented dialogs). A deeper analysis and further generalisations (*mixture of posteriors*) of the *mixture of language models* was presented in Stolcke et al. (2000). Related experiments on Maptask (Taylor et al., 1998) showed that the automatic choice of the most appropriate language model, from a set of 12 DA specific LMs (selection made using intonation modelling), can improve the speech recognition word error rate by an absolute 1%.

Machine translation is another applicative domain where DA recognition can be invaluable, since DAs can help resolving ambiguities in translating utterances. The VerbMobil project investigated machine translation in dialogue systems (Küssner, 1997; Wahlster, 2000), similarly to the work independently done by Lee et al. (1997). The use of DAs for machine translation of spoken task-oriented dialogues was also proposed in the context of the C-STAR project by Levin et al. (2003).

Automatic detection of *action items*, intended as public commitments to per-

form a defined task, is a novel research topic which shares some analogies with the DA recognition task and relies on automatically detected DA units. In the work of Purver et al. (2007), 4 task specific Action Item Dialogue Acts (AIDAs) (description, time-frame, owner and agreement) were automatically detected combining 4 independent SVM classifiers trained on: lexical, prosodic features and conventional ICSI DA tags. The automatically detected AIDAs are then rule-based parsed and summarised in order to outline the identified action items.

Disambiguating the pronoun *you* between its generic and referential use in a conversation, a task related to *action items* detection, could be useful to identify the owner of an action item (who committed to perform a given task). The SVM based system proposed by Gupta et al. (2007b), based on DAs, lexical, and part of speech features, was able to disambiguate the two uses with an accuracy of 84.4% on dyadic conversations from the Switchboard corpus. This represents a significant result, well above the baseline accuracy of 56.4%, achievable predicting always the dominant class. In particular DAs proved to be crucial for this task, reaching an accuracy of 80.92% even when used alone. Later experiments (Gupta et al., 2007a), applying a similar approach to the AMI corpus, reported an accuracy of 75.1% with the full feature setup and 71.9% using only DAs (dominant class baseline of 57.9%).

Automatically detecting when decisions are reached during a conversation is another target application for automatic DA recognition. Hsueh and Moore (2007b) used both DA unit temporal boundaries and DA labels for automatic decision detection in conversational speech. The manually annotated DA units are classified as decision making DAs or non-decision DAs using a maximum entropy classifier in conjunction with a rich set of lexical, prosodic, topical and contextual features (speaker role and DA labels). Experiments on the AMI corpus showed that decision making DAs can be detected with a precision of about 72% (66% using only contextual features such as DA labels).

Differently from written text, automatically transcribed speech lacks of a proper punctuation. It is often unpractical to process the entire raw transcription or to evaluate the resulting system on unsegmented data, thus shorter speech segments need to be defined. The temporal boundaries of automatically recognised DA units provide a principled way to segment conversational speech. For example Murray et al. (2006); Murray and Renals (2006) adopted the DA segments as the atomic

unit for automatic extractive summarisation. Features such as lexical cues, speaker activities, and term frequencies, were individually extracted from each DA unit, and Singular Value Decomposition was carried out on the resulting (DA based) feature vectors. Note that although DA segments are a good solution for automatic speech segmentation, some low-level segmentation techniques such as “Spurts” (Baron et al., 2002), continuous speech segments separated by at least half a second of silence, could represent a viable option.

More complex integrated applications based on automatic DA processing are also being investigated. For example, topic segmentation and extractive summarisation were combined in the “AMI Meeting Facilitator” system (Murray et al., 2007), a visual application focused on supporting offline meeting browsing. Dialogue acts, being exploited by both subtasks (segmentation and summarisation), offer a common ground for the whole system.

Chapter 7

Switching DBN model for joint Dialogue Act recognition

7.1 Introduction

In this chapter, we present a flexible trainable approach for the automatic recognition of Dialogue Acts in meetings, based on a switching dynamic Bayesian network model, a factored language model, and discriminative re-ranking. We present results both on the ICSI and on the AMI meeting corpora, in which we compare DA recognition accuracy on manual and automatic meeting transcriptions, and compare the effect of the different components of the overall approach. As outlined in section 6.2 the DA recognition process consists of two related tasks, DA segmentation and classification. These tasks can be performed jointly or sequentially. The joint approach, evaluating multiple segmentation and classification hypotheses, is potentially capable of better recognition performances. While the sequential technique allows an independent optimisation of the segmentation and classification tasks.

We propose an approach to DA recognition that takes advantage of both techniques by employing a joint generative infrastructure followed by a discriminative classifier. Both system components make use of supervised learning from manually annotated data, using a 15 DA class annotation scheme on AMI data and 5 broad DA classes on ICSI experiments. The joint recognition is coordinated by a switching DBN which integrates a discourse language model, six lexical and prosodic features, and two factored language models trained on the orthographic transcriptions.

The recognised sequence of DA units is then re-classified using a conditional random field DA tagger trained using the lexical content and a set of discrete features.

We have performed tagging, segmentation and recognition experiments using the joint generative approach on unseen meetings with three different modelling configurations, based on both manual and automatic speech recognition (ASR) transcriptions. We demonstrate in further experiments, that the accuracy of DA recognition using this joint approach can be further improved through discriminative post-processing.

7.2 Dialogue act recognition framework

Figure 7.1 shows our joint approach to DA recognition based on a switching DBN generative model. The observed features that are generated by this model are the words spoken by the meeting participants, together with a set of word-based prosodic features related to timing, intonation and energy. The mapping from DA labels to word sequences was modelled using a factored language model (FLM) and an interpolated FLM. The probability of observing a certain sequence of DA labels (discourse model) was represented through a simple trigram language model over DAs. The set of continuous word-based prosodic features was integrated into the recogniser using a Gaussian mixture model (GMM). The overall recognition process is actively controlled by a switching DBN which integrates information derived from words, prosodic features and language models. Section 7.3 outlines the use of an automatic speech recogniser to produce a transcription, and the extraction of the prosodic features. Sections 7.5.3 and 7.6 discuss the factored language models and the switching DBN model that underlie the DA recognition system.

7.3 Continuous features

We have used two sets of features in the DA recognition system: the transcription of the spoken words obtained using an ASR system (section 7.3.0.1) and the continuous prosodic features (section 7.3.0.2).

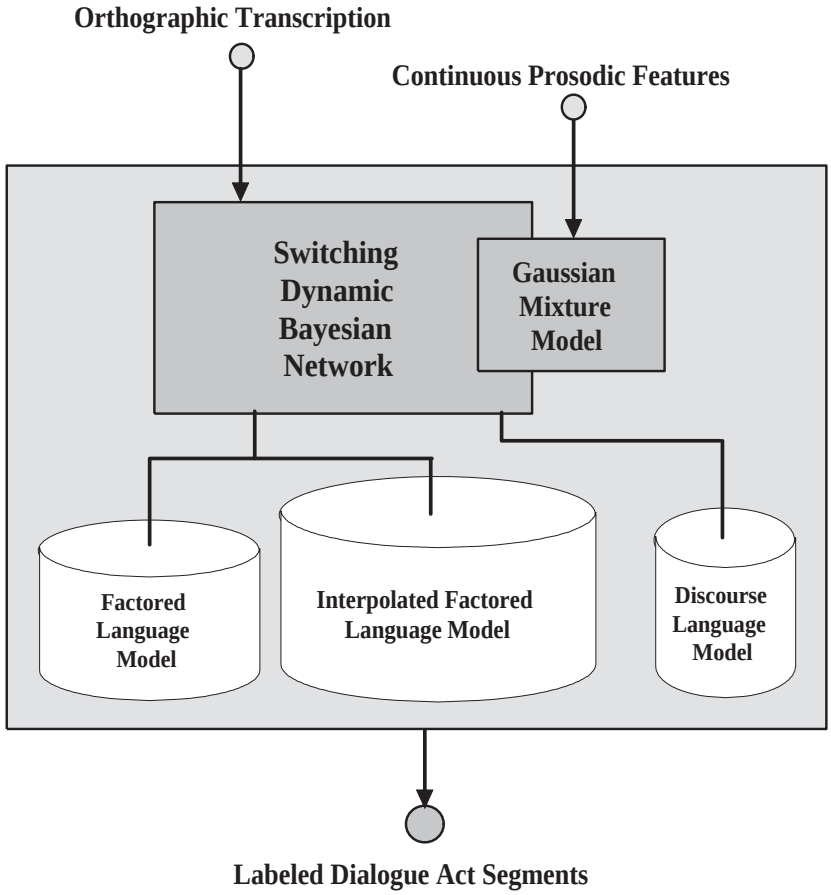


Figure 7.1: Integrated Dialogue Act recognition framework.

7.3.0.1 Speech recognition

Fully automatic DA recognition requires speech recognition. Both the ICSI and AMI corpora have been manually transcribed at the word level, as well as being processed by an ASR system, thus enabling us to assess the robustness of the DA recognition system to speech recognition errors.

Large Vocabulary Continuous Speech Recognition (LVCSR) of conversational speech is a significant research domain, and the recognition of speech in meetings has been intensively studied and evaluated in recent years [1]. Automatic transcriptions of the ICSI meeting corpus were obtained using a LVCSR system developed by the AMI-ASR team and based on: perceptual linear prediction (PLP) acoustic features, decision tree clustered crossword triphone hidden Markov models, and an

[1] NIST rich transcription meeting recognition evaluation `http://www.nist.gov/speech/`

interpolated bigram language model. The adopted system is similar to Hain et al. (2005) but acoustic models were trained using only ICSI meeting data through 4 iterations of cross-fold validation: models learned on 3/4 of the available data are used to transcribe the remaining meetings. The resulting automatic transcription achieved an overall word error rate (WER) of about 29.5%.

Automatic transcriptions of the AMI meeting corpus were obtained using the AMI-ASR system outlined in Hain et al. (2007). This LVCSR system is based on decision tree clustered crossword triphone HMMs, and a trigram language model. For the multiparty meeting domain the front end was enhanced using acoustic echo cancellation, and the perceptual linear prediction acoustic features were processed using heteroscedastic linear discriminant analysis. The acoustic feature space was normalized by speaker, using vocal tract length normalisation, and the model space was adapted using maximum likelihood linear regression. The meeting domain acoustic models were trained on the AMI corpus data. To recognize the complete corpus, a five-fold cross-validation was employed using equal splits of the corpus. Two transcription versions were generated in each case: a fully-automatic one achieved by applying the full system on automatically segmented audio files; and a semi-automatic transcription obtained from a manual segmentation into utterances. The manual system also used a simpler ASR system, in which speaker adaptation was not used. The fully automatic system resulted in an overall word error rate of about 36%; the simpler system, using manual segmentation, resulted in a WER of about 39%. In both cases the system operated on signals recorded from the close-talking microphones.

The automatic DA recognition experiments performed on the AMI corpus (section 7.7.4) compared both transcription versions. The speaker adapted “automatic segmentation” ASR output offers an overall improvement in terms of WER compared with the “reference segmentation” ASR output. However entire utterances may be deleted by the automatic acoustic segmentation, and consequently whole DA segments are irredeemably lost (section 7.7.1). Moreover, the word boundary times of the “manual segmentation” ASR output, are more accurate, compared with the reference orthographic transcription, since they cannot cross the manually annotated utterance boundaries. Accurately timed word boundaries are desirable for the extraction of prosodic features at the word level and are also required to evaluate

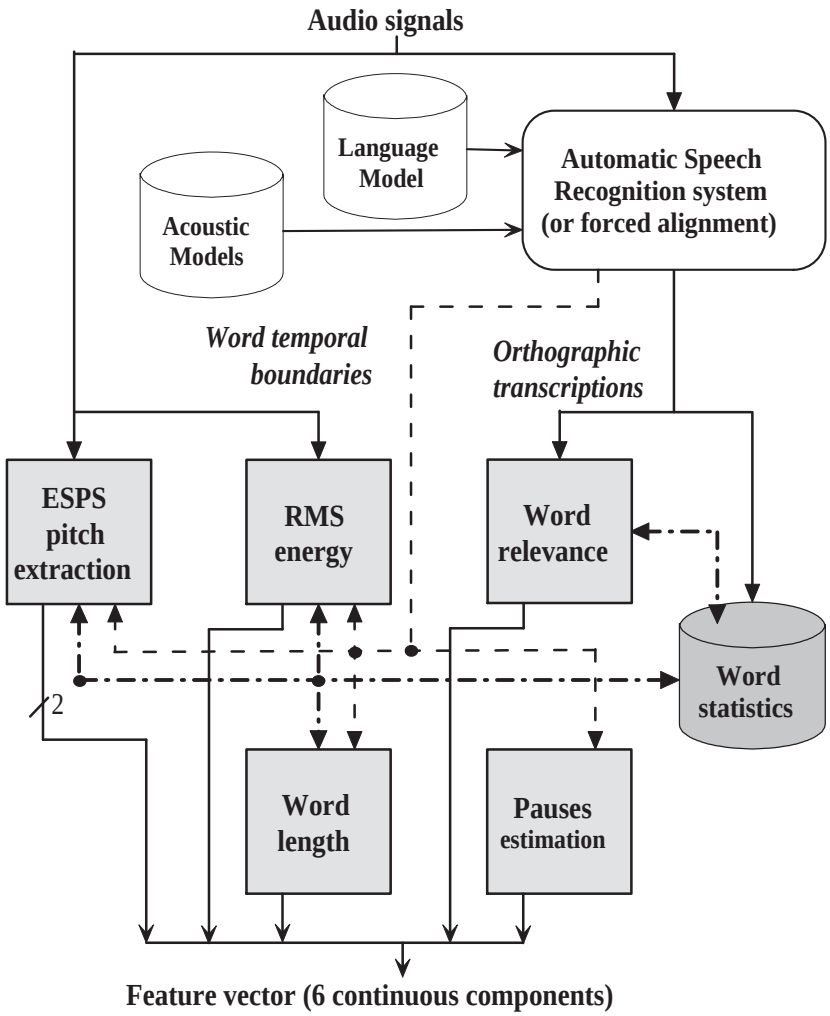


Figure 7.2: Data-flow of the automatic speech transcription and feature extraction process.

segmentation into DAs. Although both ASR versions offer valuable insights during the evaluation of our system on the AMI data, the "automatic segmentation" ASR output represents the main test condition since it does not implicate any manual intervention.

7.3.0.2 Prosodic features

Six continuous prosodic features were extracted for each word (section 3.4.2), using the audio signal and the transcription: mean and variance of the fundamental frequency (F0), mean energy, word duration, pause duration, and word informative-

ness. Figure 7.2 outlines the feature extraction process. For the reference transcription the times of word boundaries were obtained using a forced alignment against the audio. For the ASR transcriptions, the word boundary timings were output as part of the recognition process.

Mean and variance of F0 The F0 tracks were estimated using ESPS *get_f0* (Talkin, 1995) outlined in section 3.3.1.1, and the mean and variance were computed. The mean pitch was also normalised by speaker and by the average pitch for that term, with the objective of having a speaker independent measure able to highlight content words with a significant pitch shift.

RMS energy A similar normalisation technique was applied during RMS energy estimation with the aim of compensating for different channel gains (section 3.3.1.2) and to highlight emphasised words.

Word duration Word duration was "term normalised", being thus divided by the average word duration for that term, in order to highlight words which last more (or less) than the usual occurrences of that term. Therefore the resulting entity is inversely proportional to the rate of speech, neglecting estimation errors.

Word informativeness Word informativeness was estimated as the ratio between local term frequency within the current conversation and absolute term frequency across the whole meetings collection (section 3.3.3.2), thus assigning high scores to globally infrequent terms which occur frequently in the current conversation.

Pause duration Inter-word pauses were also estimated from the word boundary times. Pauses are often associated with speaker turn alternations and other relevant changes in the conversational process such as topic shifts (Stolcke et al., 1999), and it is known that they provide a valuable cue for DA segmentation (Ang et al., 2005; Zimmermann et al., 2006b).

Unit duration, pitch and energy were assigned to words which appear only once in the training set and to out-of-vocabulary words observed during testing but absent from the training set.

7.4 Discourse modelling

Generative approaches for the automatic DA recognition (or simple DA classification) usually represent the discourse structure through a probabilistic discourse model.

Generally DA classes are unevenly distributed, a clear example of this is visible in tables 3.3 and 3.4. Therefore knowledge of their prior distribution can be beneficial for the whole DA segmentation and classification process. Moreover it is reasonable to assume that the communication process tends to follow some well established patterns; for example questions are usually answered through statements, but more questions can also arise from a previous question. Each DA unit is meaningful only in the context of the entire conversation, being in a first approximation directly related to the recent conversation history: the probability to detect a new DA label is conditioned by the surrounding DA units. Note that, since generative approaches are usually based on a left to right Viterbi decoding, only the previous DA recognition history is considered.

Probabilistic discourse models are frequently adopted to account both for the prior DA class distribution and for the DA recognition history (class based probability given the previously recognised units). Different implementations of the discourse model have been proposed. For example, in a DA tagger or in a sequential DA recogniser, it is possible to include previously recognised DA labels as part of the feature vector used for the DA classification (Rosset and Lamel, 2004; Keizer and op den Akker, 2005). Elaborate language models such as polygrams were proposed (Warnke et al., 1997), but simple n-grams of DAs, being a good compromise between complexity and efficiency, represent the most frequently adopted discourse language model (Stolcke et al., 2000; Ang et al., 2005; Ji and Bilmes, 2005).

Note that DA discourse modelling has some similarities with automatic speech recognition, where conventional language models of words are used during Viterbi decoding to bridge the gap between isolated words contained in the lexicon and their (co-)occurrences in natural speech. Adapting the language model to specific communicative contexts plays a central role both in speech and DA recognition.

The discourse model adopted in our DA recognition framework consists of a

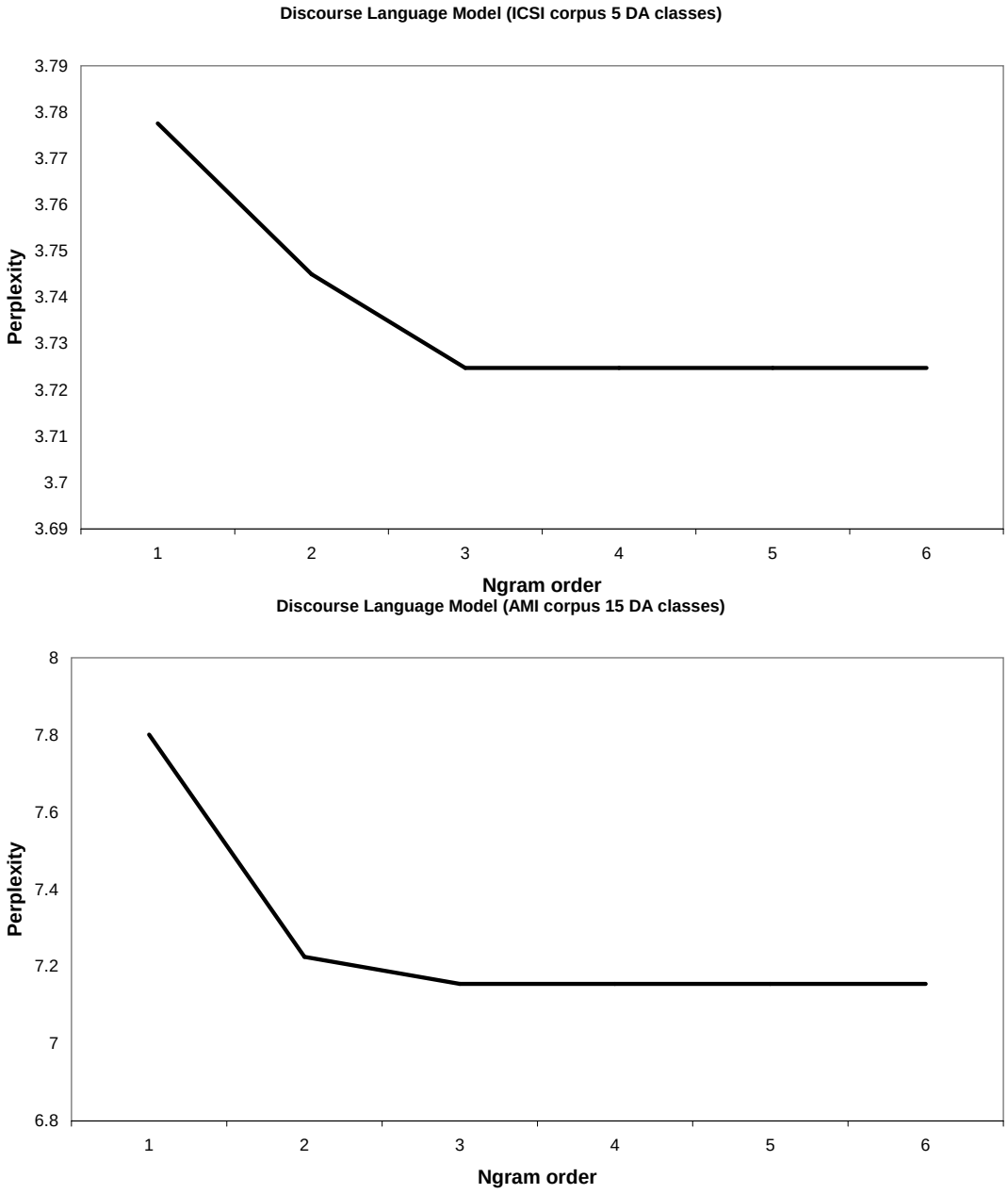


Figure 7.3: Perplexities of a unigram, bigram, ..., 6-gram discourse language model for the 5 broad DA categories of the ICSI tagging task (top) and 15 DA classes on the AMI DA classification task (bottom).

standard trigram language model over DA label sequences [2]. Sequences of manually annotated DA labels from the training set are used to train the trigram language model. The resulting discourse model is used to support the Viterbi search during testing (section 7.6). Figure 7.3 compares different discourse language model complexities (ngram orders) both on the ICSI and AMI DA classification tasks: ngram language models over DA labels are trained on the ICSI/AMI training dataset and their perplexities evaluated on unseen data. As expected, the observed perplexities are inversely proportional to the ngram order, the trigram model has the lowest perplexity, and no significant improvements are granted by higher order models (Stolcke et al., 1998).

7.5 Language modeling

As outlined in section 6.2, many different modelling approaches have been investigated to integrate lexical knowledge from the orthographic transcription in the DA classification process. Our system adopts one of the less investigated techniques: factored language models (FLMs). FLMs being more compact, flexible and elegant, represent an attractive alternative to conventional parallel ngrams (use of an individual language model for each DA class). Moreover FLM can be seen as directed graphical models and represented using a graphical paradigm (Bilmes and Kirchhoff, 2003), fitting thus well within a DBN framework.

7.5.1 Factored Language models

Conventional language models construct a joint probability distribution over word sequences, $P(w_1,\ldots,w_n)$, which is factorised as a product of conditional probabilities $P(w_t|w_{t-1},w_{t-2},\ldots,w_{t-k})$. This concept can be generalised by replacing words $w_1,\ldots,w_n$ with bundles of factors $\mathbf{v_1},\ldots,\mathbf{v_n}$, to construct a factored language model (FLM) (Bilmes and Kirchhoff, 2003). Each factor bundle, $\mathbf{v}_t \equiv \{v_t^0, v_t^1, \ldots, v_t^k\}$, is a vector whose components are factors such as word identity, part of speech tag, word stem, and enclosing dialogue act label. Conventional LMs can be interpreted as a special case of FLMs with a single factor, the actual words: $\mathbf{v}_t \equiv w_t$. Word

[2] Estimated using the SRILM toolkit, available from: `http://www.speech.sri.com/projects/srilm/`

identities are usually included in the collection of factors employed in an FLM. The smoothing and discounting techniques used for conventional LMs may be applied to FLMs, with the added flexibility of choosing which factor to drop when constructing simpler models for interpolation or backoff. Moreover, it is possible to drop more than one factor at a time and to follow multiple concurrent backoff paths using *generalised parallel backoff* (Bilmes and Kirchhoff, 2003). FLMs have an increased number of degrees of freedom, compared with conventional LMs, and it is possible to choose the factor set, the number of backoff steps, the backoff topology, and the discounting method associated to each backoff step.

We use FLMs to map word identities into DA units, and we are primarily interested in evaluating these models in terms of DA labelling accuracy (section 7.5.2), rather than perplexity. It is possible to select the optimal FLM topology automatically (Duh and Kirchhoff, 2004), and we experimented with a simple search algorithm that randomly sampled the search space. The resulting models tended to employ a large number of factors (7 or more), implying many backoff steps. These automatically discovered topologies resulted in a slightly improved DA tagging accuracy (up to 2% absolute) when compared to manually developed FLMs, but the more intricate structure requires a more elaborate DBN infrastructure and substantially increases computational cost. In order to reach a trade-off between simplicity, cost and accuracy, we decided to employ a simpler FLM topology with three factors (and two backoff steps). Although this topology was initially designed by hand, it was also discovered by the automatic search procedure (with an improved set of discounting parameters).

The FLM that we used for the DA recognition task was based on three factors: word identities w_t, the dialogue act label d_t associated to each word w_t, and the relative word position n_t in the context of the DA unit. The word sequence probability was modelled using a product of word bigrams conditioned also on word position and DA label, $P(w_t|w_{t-1}, n_t, d_t)$. The model was smoothed using two backoff steps and Kneser-Ney discounting. w_{t-1} was the first term to be dropped leading to a unigram like term, $P(w_t|n_t, d_t)$. In the case of a subsequent backoff the DA label factor d_t was the next term to be dropped, leading to $P(w_t|n_t)$. On the 5 broad DA ICSI related experiments the above described FLM was trained on the 51 meeting training set (section 3.2.2). The baseline FLM for the AMI experiments was estimated

FLM	1^{st} step	2^{nd} step	3^{rd} step	REF	ASR
$P(w_t\|w_{t-1},n_t,d_t)$	$P(w_t\|n_t,d_t)$	$P(w_t\|n_t)$		29.1	38.1
$P(w_t\|w_{t-1},p_t,d_t)$	$P(w_t\|p_t,d_t)$	$P(w_t\|p_t)$		36.5	45.1
$P(w_t\|w_{t-1},m_t,d_t)$	$P(w_t\|m_t,d_t)$	$P(w_t\|m_t)$		31.2	39.8
$P(w_t\|w_{t-1},n_t,m_t,d_t)$	$P(w_t\|n_t,m_t,d_t)$	$P(w_t\|n_t,d_t)$	$P(w_t\|n_t)$	31.5	39.6
$P(w_t\|w_{t-1},n_t,p_t,m_t,d_t)$	$P(w_t\|n_t,p_t,m_t,d_t)$	$P(w_t\|n_t,p_t,d_t)$	$P(w_t\|p_t)$	37.1	45.3
Integrated system: $P(w_t\|w_{t-1},n_t,d_t)$ FLM + DBN infrastructure				24.0	34.5

Table 7.1: DA tagging error rate (%) on the ICSI meeting corpus using the Factored Language Model alone; results have been reported on 5 different FLM setups and compared with the fully integrated FLM+DBN infrastructure.

using the training subset of the AMI scenario meeting data outlined in section 3.2.3 (470 000 words and a dictionary of about 9 000 unique terms).

7.5.2 Classification performances of a FLM based DA tagger

In order to benchmark different FLM topologies, instead of comparing their perplexities, we have defined a simplified *DA tagging* task. We compared different FLMs by measuring their ability to assign the correct DA label to unseen DA units. This preliminary evaluation was conducted by enhancing the FLM module of the SRILM toolkit (Stolcke, 2002) with a simple decoder, able to label each DA unit (sentence) with the most likely DA tag (factor label from a list of possible options). Being interested in benchmarking different FLMs, and needing an unbiased comparison of the candidate topologies, we decided to disregard the classification history in these preliminary experiments. However a trigram discourse language model was included in all the switching DBN based DA recognition experiments reported in section 7.7.

As shown in table 7.1, the 3 factors FLM $P(w_t|w_{t-1},n_t,d_t)$, after training on the 51 ICSI meeting training set, was able to perform DA labeling on the 11 ICSI test

FLM	1^{st} step	2^{nd} step	3^{rd} step	REF	ASR_AS
$P(w_t\|w_{t-1},n_t,d_t)$	$P(w_t\|n_t,d_t)$	$P(w_t\|n_t)$		47.7	59.7
$P(w_t\|w_{t-1},p_t,d_t)$	$P(w_t\|p_t,d_t)$	$P(w_t\|p_t)$		52.6	63.1
$P(w_t\|w_{t-1},m_t,d_t)$	$P(w_t\|m_t,d_t)$	$P(w_t\|m_t)$		49.3	61.6
$P(w_t\|w_{t-1},n_t,m_t,d_t)$	$P(w_t\|n_t,m_t,d_t)$	$P(w_t\|n_t,d_t)$	$P(w_t\|n_t)$	48.9	61.5
$P(w_t\|w_{t-1},n_t,p_t,m_t,d_t)$	$P(w_t\|n_t,p_t,m_t,d_t)$	$P(w_t\|n_t,p_t,d_t)$	$P(w_t\|p_t)$	53.5	65.4
Integrated system: $P(w_t\|w_{t-1},n_t,d_t)$ FLM + DBN infrastructure				40.9	52.7

Table 7.2: DA tagging error rate (%) on the AMI meeting corpus using the Factored Language Model alone; results have been reported on 5 different FLM setups and compared with the fully integrated FLM+DBN infrastructure.

meetings with a classification error rate of 29.1% using reference transcriptions and 38.1% using automatic transcriptions. Replacing the word position factor n_t with part-of-speech tags p_t (automatically labelled using a POS tagger trained on Broadcast News data) the error rate on manual transcriptions rised to 36.5%. Building the model $p(w_t \mid w_{t-1}, m_t, d_t)$, where m_t represents the information about the meeting type [3], the error rate fell to 31.2%. An error rate of 31.5% could be achieved integrating both n_t and m_t into $P(w_t|w_{t-1},n_t,m_t,d_t)$. Finally a three backoff steps FLM which includes all the three factors n_t, p_t and m_t, scored a significatively higher classification error rate of 37.1%. All the FLM setups showed similar behaviours on ASR automatic transcriptions. Note that $P(w_t|w_{t-1},n_t,d_t)$ outperformed the other FLM candidates both on reference manual transcriptions and ASR output, and a further improvement was granted by the fully integrated system (based on a switching DBN, a FLM, and a discourse model). Similar results were achieved on the AMI meeting corpus: $P(w_t|w_{t-1},n_t,d_t)$ was the best performing topology, followed by $P(w_t|w_{t-1},n_t,m_t,d_t)$ and $P(w_t|w_{t-1},m_t,d_t)$. Word relative position n_t and meeting

[3] ICSI meeting series (database issues, network services, meeting recorder project, SRI collaboration, etc.) or including information about the AMI instrumented meeting room (Edinburgh, IDIAP, or TNO) along with meeting intent (kick-off, functional design, conceptual design, or detailed design meeting).

type m_t were the most effective factors in both ICSI and AMI DA classification experiments. A qualitative comparison [4] between table 7.1 and 7.2 suggested that the automatic classification of the 15 unevenly distributed AMI DA classes is more challenging than recognising the 5 generic DA categories employed during the ICSI experiments.

7.5.3 Interpolated Factored Language models

FLMs with the same topology may be interpolated, similarly to word-based n-grams. This enables the construction of combined models, whose component FLMs are trained using different data resources. For example on the 15 DA AMI experiments (section 7.7.4), we built FLMs for DA recognition using the ICSI meetings corpus and the Fisher corpus of conversational telephone speech, in addition to an FLM built on the target AMI corpus, integrating them into a single interpolated factored language model.

The AMI meetings corpus has a size of 0.97 million words in total, with about 0.47 million words in our training set of 98 meetings. The ICSI corpus (section 3.2.2), which is from a similar domain, contains 0.74 million words. The Fisher corpus (section 3.2.4), which is based on two party telephone conversations is much larger, containing 10.62 million words. Building an interpolated FLM from these data sources, enriches the baseline FLM trained on AMI meetings only, by extending the vocabulary and thus reducing the out-of-vocabulary, and by improving the n-gram counts with word sequences that are not observed in the AMI training dataset alone. However, neither the ICSI or Fisher corpora are annotated using the AMI DA annotation scheme. (The ICSI corpus has been annotated for DAs, but using a different and incompatible scheme.) In the absence of useful DA annotations, both the ICSI and FISHER corpora were duplicated 15 times when training the FLMs, labeling every sentence with all the 15 possible DA labels in the AMI DA annotation scheme. FLMs trained on artificially duplicated data are obviously not discriminative in a DA classification task, but they are able to enhance the dictionary and n-gram counts of the resulting interpolated FLM.

Note that a similar procedure has been applied to the ICSI 5 broad DA recog-

[4] ICSI and AMI DA annotation schemes are incompatible, thus a principled quantitative comparison is not possible.

nition experiments (section 7.7.3): AMI and FISHER data are duplicated 5 times, one for each DA class, and used to train two additional FLMs; the resulting non DA discriminative FLMs are then interpolated with the baseline FLM trained only on ICSI data.

As will be discussed in section 7.7 the use of an interpolated FLM provides an improvement in DA segmentation at the price of slightly reduced DA classification accuracy. To address this, we conducted experiments with a hybrid approach in which the baseline FLM trained on the AMI data (ICSI data for the experiments reported in section 7.7.3) is combined with an interpolated FLM at the sequence decoding level by maximising the product of the joint probabilities associated to the two concurrent FLMs.

7.6 DBN based framework

In a DA recognition system, segmentation and classification are strongly related—the output of the DA classifier is dependent on the optimal placement of the DA unit boundaries, and the placement of the DA boundaries depends on the labels assigned to the DAs. In our approach, we treat the segmentation and classification problems jointly and the process is coordinated by a switching DBN model (section 2.5), implemented using the Graphical Model ToolKit (GMTK) outlined in section 2.7.

Figure 7.4 depicts the switching DBN model (Dielmann and Renals, 2007b). The transcribed words are represented as the sequence of discrete observable nodes $W_0, \ldots, W_{t-1}, W_t$. The FLM and interpolated FLM outlined in the previous sections are depicted using dotted arcs, and each word is observed twice: once for the baseline FLM and once for the interpolated FLM. The relative position of each word W_t into the current DA unit DA_t^0 is represented by the discrete node N_t. N_t relies on a bounded word counter C_t, which is incremented at every word encountered in the current DA unit. After each block of 5 words, C_t is reset to zero and N_t is incremented, thus indicating to which "block of five words" the current word W_t belongs to:

$$\begin{aligned} &\text{if } C_{t-1} < 4: \quad C_t := C_{t-1} + 1 \\ &\text{if } C_{t-1} = 4: \quad C_t := 0 \qquad N_t := N_{t-1} + 1 \quad . \end{aligned} \tag{7.1}$$

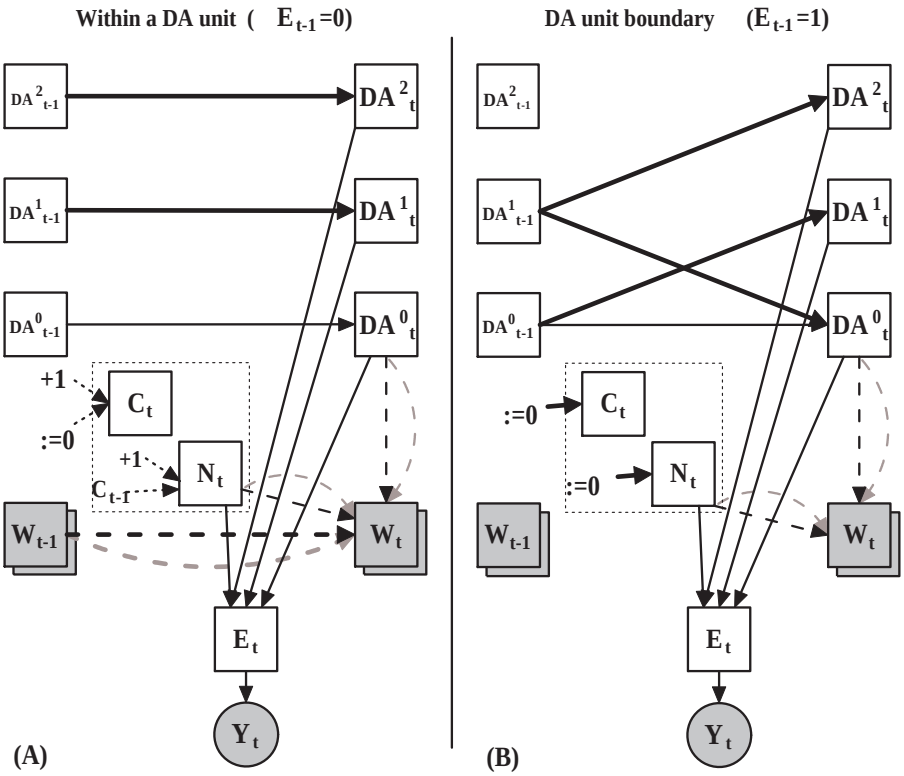


Figure 7.4: Switching dynamic Bayesian network model for the joint dialogue act recognition task: *Intra-DA* topology adopted within a DA unit (A); *Inter-DA* topology used at DA boundaries (B). The model switches between the two operating conditions (topologies) according to the state of the DA boundary detector node E. Square nodes represent discrete random variables, round nodes are continuous variables. Shaded nodes represent observable features, unshaded nodes are hidden variables. Plain arcs visually encode statistical dependences between random variables and dotted arcs highlight the dependences implied by FLMs.

The final length of an automatically detected DA unit is not known a priori, and is only available at the end of the DA recognition process, therefore it is impractical to estimate word position features normalised for DA length.

The DA recognition history is represented by the current and the two previous DA labelling hypotheses, DA_t^0, DA_t^1 and DA_t^2. This history is needed by the DA boundary detector, the hidden binary variable E_t. E_t is the principal switching variable in the model, switching from zero to one when a boundary between two disjoint DA units is detected. In the absence of a DA boundary ($E_{t-1} = 0$) the DBN assumes the *intra-DA* topology shown in figure 7.4(A); when a boundary is likely to be present ($E_{t-1} = 1$) the model adopts the alternative *inter-DA* topology depicted in figure 7.4(B).

The dependency of the observable prosodic feature vectors Y_t on E_t is modelled using a Gaussian mixture model (GMM) with n components:

$$P(Y_t = y \mid E_t = i) = \sum_{j=1}^{n} C(i,j)\mathcal{N}(y; \mu_{i,j}, \Sigma_{i,j}) \tag{7.2}$$

where $\mathcal{N}(y; \mu_{i,j}, \Sigma_{i,j})$ is a Gaussian density with mean $\mu_{i,j}$ and covariance $\Sigma_{i,j}$, evaluated at y. $C(i,j)$ is the conditional prior weight of each mixture component j, and the optimal number of mixture components n for each state $i = [0,1]$ is automatically selected during training (Bilmes and Zweig, 2002). The GMM relates the six-dimensional prosodic features to the two discrete states of E_t, thus helping to predict the DA segmentation.

The cardinalities of the discrete random variables reflect the function they serve in the model, thus: $|E_t| = 2$, $|C_t| = 5$, $|DA_t^0| = |DA_t^1| = |DA_t^2| = 15$ on AMI experiments, $|DA_t^0| = |DA_t^1| = |DA_t^2| = 5$ on ICSI experiments, and W_t has as many states as the number of words in the dictionary. Since the vast majority of the DA units have fewer than 75 words, the word block counter cardinality has been constrained to $|N_t| = 15$.

The intra DA topology used within a DA unit (figure 7.4(A)) accumulates the joint probability for a sequence of $k+1$ words $W_{t-k},\ldots,W_t$ as the product of a FLM and a weighted interpolated FLM given the current DA label hypothesis DA_t^0 and the deterministic counter nodes N_t and C_t. The two language model probabilities (FLM and interpolated FLM) are combined by using an equally weighted stream

weighting combination:

$$P(W_{t-k},\ldots,W_t \mid DA^0) = \prod_{i=t-k}^{t} \{P_{IFLM}(W_i \mid W_{i-1}, N_i, DA^0) \cdot \cdot P_{FLM}(W_i \mid W_{i-1}, N_i, DA^0)\} \quad (7.3)$$

where $P(W_{t-k},\ldots,W_t \mid DA^0)$ represents the joint probability for the observed utterance W_{t-k},...,W_t, given the current DA classification hypothesis DA^0; P_{FLM} and P_{IFLM} are the probabilities respectively provided by the baseline and the interpolated FLMs.

The absence of a DA boundary implies that the DA recognition history remains unaltered, hence the content of DA^1_{t-1} needs to be cloned into DA^1_t and similarly $DA^2_t := DA^2_{t-1}$. Since the word sequence W_{t-k},...,W_t was generated by the same DA unit with label DA^0_t, and no DA boundaries were spotted between time $t-k$ and time t, it follows that $DA^j_{t-k} = \ldots = DA^j_{t-1} = DA^j_t$ for $j = [0,2]$.

If a DA boundary is hypothesised ($E_{t-1} = 1$), then the model switches to the inter DA topology (figure 7.4(B)), which integrates the probability from the 3-gram discourse LM into the overall recognition process and starts the evaluation of a new DA unit, reinitialising the counter nodes: $C_t = 0$, $N_t = 0$. The DA recognition history is updated and a new set of DA classification hypotheses DA^0_t, for the next DA unit beginning with W_t, is generated following the 3-gram discourse language model $P(DA^0_t \mid DA^1_{t-1}, DA^2_{t-1})$.

When $t = 0$ a slightly modified intra DA topology ($E_{-1} = 0$) needs to be adopted: having both the DA recognition history and the counter nodes forcefully initialised to zero ($DA^1_0 = DA^2_0 = 0$, $C_0 = 0$, $N_0 = 0$).

Segmentation and classification are carried out concurrently. The classification process accounts for the joint probability of the transcription $W_{t-k},\ldots,W_t$ accumulated by the two concurrent FLMs given the current classification hypothesis DA^0_t, the probability of DA^0_t given the two previously recognised DA units, and the segmentation hypothesis (a DA unit starting at time $t-k$ and ending at time t). Several alternative segmentation hypotheses are generated, with the probability of each segmentation combining the likelihood of generating the observed prosodic feature vectors Y_t and the likelihood of the DA unit generating the observed words

$W_{t-k},\ldots,W_t$. A pruned Viterbi decoding is used to find the most likely sequence of labeled DA segments [5].

Since this approach cannot generate a DA segmentation without an associated DA labeling hypothesis, the segmentation accuracy is assessed by ignoring the recognised DA labels. Classification of the DA units for a reference segmentation can be achieved by constraining the state of the boundary detector nodes E.

7.7 Experimental results

Evaluation methodologies and metrics for automatic DA tagging, segmentation and recognition tasks are outlined in the following section 7.7.1. Experimental results on ICSI and AMI data, using the switching DBN based infrastructure and 3 FLM configurations, are reported respectively in section 7.7.3 and 7.7.4. Discriminative re-classification of the automatic AMI DA segmentation will be discussed in section 7.8.

7.7.1 Performance evaluation metrics

DA tagging accuracy can be easily evaluated by scoring the automatic DA classification output on a test set against the corresponding reference DA annotation. The percentage of correctly classified DA units, or its complement the Classification Error Rate, is a standard metric for the DA tagging task, along with class-based precision and recall measures (Lesch et al., 2005a).

The evaluation of DA segmentation accuracy is less straightforward. The concept of a "correct" DA segmentation is not unequivocally defined, since it may be in terms of the overall sequence of DA units, or may demand precise timing of the DA boundaries. Moreover a segmentation metric may be expressed and normalised in terms of DA units, DA boundaries or words. A number of different metrics have been proposed, each offering a different perspective on the task of DA segmentation. In this thesis we report our results using four previously defined metrics: the NIST Sentence like Unit (NIST-SU), Strict, and Boundary metrics (Ang et al.,

[5] The decoding runtime for this model using 15 DA classes is about 10 times slower than realtime on a 3Ghz P4 equipped with 1Gb of RAM, however these computational costs scale exponentially with the number of target DA labels.

	Normalised by:		
	DA boundaries	Words	DA units
Tolerant: 1 matching boundary	NIST-SU *NIST-SU*	Boundary	
Rigorous: 2 matching boundaries		Strict *Strict*	DSER *DER*

Table 7.3: DA segmentation and *recognition* evaluation metrics.

2005), and the DA Segmentation Error Rate (DSER) metric (Zimmermann et al., 2006a,b). These metrics are summarised in table 7.3.

According to the Strict and DSER metrics a DA unit has been correctly detected only when both boundaries are correctly located and no other boundaries fall within the detected unit; the NIST-SU and Boundary metrics focus on individual boundaries, rather than on DA units, and are thus more tolerant. The NIST-SU metric scores the sum of missed DA boundaries and false-alarms against the number of reference DA boundaries. In case of a high number of insertions (false-alarms) the NIST-SU metric can assume values well above 100% (Zimmermann et al., 2006b). The Boundary metric has the same numerator as the NIST-SU metric (missed boundaries + insertions) but is normalised by the total number of non-boundaries in the reference, which is equivalent to number of reference words. Since there are usually many more reference words than segmentation errors, this metric tends to be skewed toward very low error rates. The DSER metric is the complement of the percentage of correctly detected DA units; similarly the Strict metric can be defined as the percentage of words belonging to incorrectly segmented units. The Strict metric is a severe metric heavily influenced by the length of DA units in terms of words.

Since the DA recognition task combines segmentation and tagging, it is possible to translate most of the segmentation metrics into recognition metrics by requiring that the detected DA unit labels match the reference annotation. Therefore the NIST-SU, Strict, and DSER (usually referred as DA Error Rate or DER in the recognition task) metrics can be easily adapted to the recognition task by adding the constraint that wrongly labeled units will be scored as errors even if their bound-

aries are a perfect match. This added requirement implies that these recognition metrics will result in error rates at least as great as their segmentation counterparts. The Boundary segmentation metric is an exception, since it is translated into the Lenient recognition metric (Ang et al., 2005), which is defined as the percentage of correctly classified words independent of the segmentation. Since it is focused exclusively on tagging accuracy, this metric should be regarded as a DA classification metric rather than a genuine recognition metric.

The reference DA annotation is produced in terms of the manually transcribed word sequence. When processing ASR output, the DA tags will be applied to a different word sequence, owing to ASR errors. Since a manual re-annotation of the ASR output would be extremely expensive, we have adopted the evaluation scheme proposed by Ang et al. (2005): ASR words are mapped into the manually annotated segments according to their midpoint $0.5 * (word\_start\_time + word\_end\_time)$, thus inheriting their reference DA labels. Because of ASR deletions and the time-based alignment, several DA units will be empty. As we have adopted a word-based approach, these lost segments cannot be successfully recognised and will be reported as errors by every segmentation/recognition metric. Conversely on a pure DA tagging evaluation task, empty segments will be scored as if they were tagged with a randomly drawn label, thus reducing the biasing effect of words and utterances deleted by the ASR system.

7.7.2 Experimental setup

We have used the switching DBN model for tagging, segmentation, and recognition of DAs in the ICSI and AMI meeting corpora, using the three language model configurations described in section 7.5.3: FLM, interpolated FLM, and a hybrid in which the interpolated FLM is focused on segmentation and the baseline FLM is focused on tagging. These experiments extend our previously published results in which an early version of the switching DBN model, without the use of interpolated FLMs, was used for DA recognition on the ICSI meetings corpus (Dielmann and Renals, 2007c), and experiments on the AMI corpus using manual transcriptions only (Dielmann and Renals, 2007b). Our initial experiments, applying the complete framework to the 5 DA ICSI task (section 7.7.3), validates the methodology on an established task, forming the base for our investigations on the novel 15 DA

AMI task (section 7.7.4). In order to validate the 6 continuous features proposed in section 7.3.0.2, experiments using individual features classes and the full feature set have been conducted on the 5 DA ICSI task.

7.7.3 Numerical results on the ICSI corpus

All the experiments on the ICSI corpus were performed using the five DA categories and the data sets described in section 3.2.2. DA tagging, segmentation and recognition results are reported both on the reference orthographic transcription and using the output of automatic speech recognition. As outlined in section 7.3.0.1, the automatic transcription of the ICSI corpus was provided by the AMI ASR team and generated through an ASR system similar to the one outlined in Hain et al. (2005) (word error rate of about 29%).

Although our system is primarily targeted on the DA recognition task intended as joint segmentation and classification, it is possible to provide the ground truth segmentation and evaluate the DA tagger alone. DA tagging classification error rates reported in table 7.4 show that the percentage of incorrectly labeled units is about 24.0% on reference transcriptions and about 34.5% on ASR output. The classification procedure is exclusively based on the lexical information (through the FLM) and on the DA language model; prosodic related features are used only for segmentation and overall recognition purposes. Comparing these results with those reported in table 7.1, we can deduce that the introduction of a trigram discourse model resulted in an absolute improvement included between 3% (on automatic transcriptions) and 5% (on manual transcriptions).

Table 7.4 also shows the segmentation and recognition results on five different setups. Results are reported using all the evaluation metrics outlined in section 7.7.1. Note that all the eight adopted metrics are “error rates”, thus lower numbers correspond to better performances. The proposed setups differ only in the information used to detect DA boundaries: the *Lexical* setup makes no use of continuous features (node Y was removed from the DBN), the *Prosody* setup uses only five out of six features (excluding pauses), the *Pause* setup uses the pause information but not the other continuous features, the *All (REF)* and *All (ASR)* configurations exploit the full feature set. *All (REF)* reports the results achieved by training and evaluating the DA recogniser on manually annotated orthographic transcrip-

tions, whenever in *All (ASR)* the system was developed and tested on automatic transcriptions. Therefore in the later experiment the combination of ASR and DA recogniser constitutes a fully automatic approach, since manual annotations are not needed. Significance testing using the Matched Pair Sentence Segment Word Error (MAPSSWE) test (Pallett et al., 1990; Jurafsky and Martin, 2008) showed that *Lexical* and *Prosody* setups are not significantly different, and that all the other systems are significantly different at level $p = 0.001$.

Note that the *Lexical* setup makes use of the lexical information just for DA classification purposes. Boundary detection is estimated from the current DA label, the DA history and the word block counter. Therefore this setup and the lexically based systems investigated in Zimmermann et al. (2006a) cannot be directly compared.

The adoption of prosodic and word related features made in the *Prosody* setup presents a conflicting behaviour: NIST-SU, strict and boundary metrics show an improvement over the baseline setup; while DSER, DER, and lenient metrics move toward higher error rates. The *Pause* setup shows a clear improvement over the baseline approach under all the evaluation metrics, and proves its strength over the *Prosody* setup highlighting the importance of pause related information on the segmentation task.

The fully integrated approach (*All-REF*) is the most accurate model. The error rates are similar to the NIST-SU segmentation error rate (34.4%) and the lenient recognition error rate (19.6%) of the two step recogniser presented by Ang et al. (2005) (section 6.2). This result suggests that, even if the two competing systems have similar segmentation performances, and the maximum entropy based DA classifier (about 80% correct classification (Ang et al., 2005)) seems to be more powerful than our generative approach, the joint segmenter+classifier framework is potentially able to outperform a sequential framework.

This is even more evident with the fully automatic ASR based system (*All-ASR*) which provides a relevant improvement if compared [6] to the sequential approach outlined in Ang et al. (2005) (lenient recognition error rate of 25.1%). In the sequential approach the DA classifier is able to process only one segmentation hypothesis, whereas in the joint approach multiple segmentation hypotheses are taken

[6] A strict direct comparison to the ASR condition of Ang et al. (2005) will require their exact automatic transcription, which is not available.

Task	Metric	LEXICAL	PROSODY	PAUSE	ALL-REF	ALL-ASR
Tag.	100 - %Correct	24.0			24.0	34.5
S	NIST-SU	93.7	83.4	48.0	**35.6**	**43.6**
E	DSER	83.6	90.7	51.2	48.9	58.2
G	STRICT	87.4	85.8	66.4	56.5	63.5
M.	BOUNDARY	14.5	12.9	7.4	5.5	7.3
R	NIST-SU	104.1	93.8	68.5	56.8	69.6
E	DER	86.7	92.1	62.9	61.4	72.1
C.	STRICT	89.1	87.6	72.5	64.7	**72.5**
	LENIENT	20.7	22.0	19.5	**19.7**	**22.0**

Table 7.4: DA tagging, segmentation and recognition error rates (%) on the ICSI meeting corpus; results using the baseline FLM are reported on 3 individual feature setups (lexical, prosody and inter-word pauses); experiments using a combined feature setup are reported both on reference (ALL-REF) and automatic transcriptions (ALL-ASR).

		Reference transcription		
Task	Metric	FLM	iFLM	Hybrid
TAG.	100 - %Correct	**24.0**	38.8	25.2
S	NIST-SU	35.6	**30.5**	32.0
E	DSER	48.9	**27.9**	27.8
G	Strict	56.5	**50.3**	52.3
M.	Boundary	5.5	**4.7**	4.9
R	NIST-SU	56.8	67.9	**59.5**
E	DER	61.4	57.9	**47.4**
C.	Strict	64.7	66.4	**62.7**
	Lenient	19.7	30.3	**20.9**

Table 7.5: DA tagging, segmentation and recognition error rates (%) on the ICSI meeting corpus using a dictionary of 5 broad DA classes; results are reported on 3 different FLM setups (baseline FLM, interpolated FLM, and hybrid FLM+iFLM) using reference manual transcriptions.

in account by the DA tagger. The final choice between multiple candidates will be carried out by taking the most likely sequence of DA units, intended as the optimal combination of DA boundaries and DA labels.

Further results obtained comparing three language model configurations are reported in table 7.5: the baseline FLM model (*All-REF* column of table 7.4); a novel weighted interpolated FLM trained on ICSI, AMI and Fisher data (AMI and Fisher were duplicated 5 times, one for each DA class); and a *hybrid* combination of the two FLMs. These experiments indicate that the baseline FLM offers the best tagging performance; adoption of an interpolated FLM improves the segmentation accuracy at the cost of tagging. An effective trade-off between DA tagging and segmentation, required for DA recognition, was obtained using the *hybrid* configuration (baseline FLM and interpolated FLM used in conjunction). The same behaviour can be ob-

		Reference transcription		
Task	Metric	FLM	iFLM	Hybrid
S	NIST-SU	32.4	14.8	14.9
E	DSER	45.0	17.1	16.2
G	Strict	47.2	27.4	27.0
M.	Boundary	4.7	2.2	2.2
R	NIST-SU	50.2	36.8	30.9
E	DER	55.4	36.1	29.2
C.	Strict	33.2	32.5	30.7
	Lenient	14.2	7.8	6.0

Table 7.6: DA segmentation and recognition error rates (%) on the training set of the ICSI meeting corpus using a dictionary of 5 broad DA classes; results are reported on 3 different FLM setups (baseline FLM, interpolated FLM, and hybrid FLM+iFLM) using reference manual transcriptions.

served on the ICSI training dataset (table 7.6): the iFLM setup focuses on detecting an accurate segmentation and the hybrid approach optimises the overall recognition performances.

MAPSSWE significance testing reported a significant difference at level $p = 0.001$ between *iFLM* and the baseline *FLM* system, *iFLM* and *hybrid*; and a difference at level $p = 0.01$ between *hybrid* and *FLM*. The automatic recognition outputs, provided by the three systems reported in table 7.5, agree with the reference DA annotation according to a Kappa of $k = 0.6$ (FLM), $k = 0.56$ (iFLM), and $k = 0.63$ (hybrid). The human upper bound performances are given by the inter-annotator agreement, which on this 5 DA recognition task is about $k = 0.8$ (section 3.2.2).

Our results applying the switching DBN model to the ICSI task compare favorably to the combined joint approach of Zimmermann et al. (2006b). Although for tagging the FLM is less accurate than a discriminative DA classifier (Ang et al., 2005), the situation is inverted on the DA segmentation task (Zimmermann et al.,

2006b), thanks to the added capability to include additional in-domain data by adopting an interpolated FLM. Our joint recognition experiments suggest that these two effects can be carefully balanced (hybrid approach), leading to a competitive DA recogniser which performs well in comparison with the state of the art (Zimmermann et al., 2006b).

7.7.4 Numerical results on the AMI corpus

We have used the switching DBN model for tagging, segmentation, and recognition of DAs in the AMI meeting corpus (from a dictionary of 15 DAs), using three language model configurations described in section 7.5.3: FLM, interpolated FLM, and a hybrid in which the interpolated FLM is focused on segmentation and the baseline FLM is focused on tagging. Each of these systems was run on three transcription conditions: manual reference transcription, ASR with manual utterance segmentation, and ASR with automatic utterance segmentation. As discussed in chapter 3, the AMI meeting corpus uses a set of fifteen DA classes, in contrast to the five broad DA classes used on the ICSI corpus, thus results for the two corpora are not directly comparable.

Error rates for the DA tagging, segmentation and recognition tasks, using the three system configurations and the three transcription conditions are shown in table 7.7. The three system configurations are as follows:

- *FLM*: simple FLM trained only on the AMI training set;
- *iFLM*: weighted interpolated FLM trained on AMI (relative combination weight of about 58.5%), ICSI (2.7%) and FISHER (38.8%) conversational data;
- *Hybrid*: *iFLM* and *FLM* combined at the decoding level.

These three systems were each run on three transcription conditions, described in section 7.3.0.1:

- *Manual* Hand transcription (WER: 0%);
- *ASR_AS* ASR with automatic segmentation: fully automatic system from ASR preprocessing up to DA segmentation and recognition (WER: 36%; 12.8% of DAs lost due to ASR deletions);

- *ASR_MS* ASR with manual segmentation: non-speaker adapted ASR with manual utterance segmentation (WER: 39%; 5.8% of DAs lost due to ASR deletions).

Although *ASR_MS* has a higher word error rate, the manual segmentation results in fewer complete DAs being deleted. Most of the deleted DA segments are very short, typically backchannels or fragments; an example of this is visible at the bottom of figure 7.6. Note that all the three proposed systems (*FLM*, *iFLM*, and *Hybrid*) showed significant differences at level $p = 0.001$ according to the MAPSSWE test.

The *FLM* system has a classification error rate of about 10% absolute lower than the *iFLM* system for the tagging task, which uses a predefined segmentation. This is to be expected, since the additional data sources used in the *iFLM* system, the Fisher and ICSI corpora, do not have DA tags corresponding to the AMI scheme (section 7.5.3). Thus although these additional data sources extend the vocabulary and n-gram counts, they are unable to provide information to help discriminate between DA classes. The trigram discourse model contributes to these results by about 7.0% absolute: DA tagging experiments using the *FLM* system without the discourse trigram, resulted in classification error rates of 47.7%, 57.5% and 59.7% respectively for the *manual*, *ASR_MS* and *ASR_AS* transcriptions.

Precision and recall of DA tagging is shown by class in figure 7.5. This graph indicates that DA tagging accuracy is influenced by the imbalanced distribution of DA labels. Not surprisingly the classifier performs better on the two most frequent classes, *inform* and *backchannel*. However very infrequent classes such as *be-positive* and *offer* have good recall and precision scores, suggesting that even if rare they can be well modelled and discriminated.

For the DA segmentation task, table 7.7 indicates that the *iFLM* system results in much lower errors, by a factor of three, compared with the basic *FLM* approach. In this case the reduced discrimination of the *iFLM* system is outweighed by the extended dictionary and larger language model, obtained from the additional ICSI and Fisher corpora.

Since DA recognition needs both accurate segmentation and classification, we combined the *FLM* and *iFLM*, resulting in a hybrid approach which combines the two models at the decoding level. The segmentation error rates of the *hybrid* system are slightly higher than those provided by the *iFLM* approach, and the tagging error

		Reference transcription			ASR manual segmentation			ASR automatic segmentation		
Task	Metric	FLM	iFLM	Hybrid	FLM	iFLM	Hybrid	FLM	iFLM	Hybrid
TAG.	100 - %Correct	**40.9**	51.4	42.8	**50.7**	61.2	53.0	**52.7**	61.9	54.8
S	NIST-SU	70.7	**20.4**	25.6	77.6	**26.5**	34.1	102.6	**30.7**	34.0
E	DSER	78.0	**12.8**	17.0	85.5	**17.0**	22.8	94.2	**23.2**	25.8
G	Strict	74.4	**28.5**	36.9	81.8	**29.4**	39.5	91.5	**26.9**	33.7
M.	Boundary	10.8	**3.1**	3.9	12.8	**4.4**	5.6	16.7	**5.0**	5.5
R	NIST-SU	93.1	73.6	**71.3**	98.3	85.3	**85.9**	114.8	84.0	**81.2**
E	DER	85.5	57.0	**51.9**	91.7	67.0	**62.5**	96.5	68.6	**64.1**
C.	Strict	83.2	64.4	**62.1**	89.2	70.7	**68.5**	94.5	68.3	**64.7**
	Lenient	40.9	51.8	**42.2**	43.8	59.0	**48.3**	43.4	57.1	**46.9**

Table 7.7: DA tagging, segmentation and recognition error rates (%) on the AMI meeting corpus; results are reported on 3 different FLM setups (baseline FLM, interpolated FLM, and hybrid FLM+iFLM) both on reference manual transcriptions and on 2 ASR outputs.

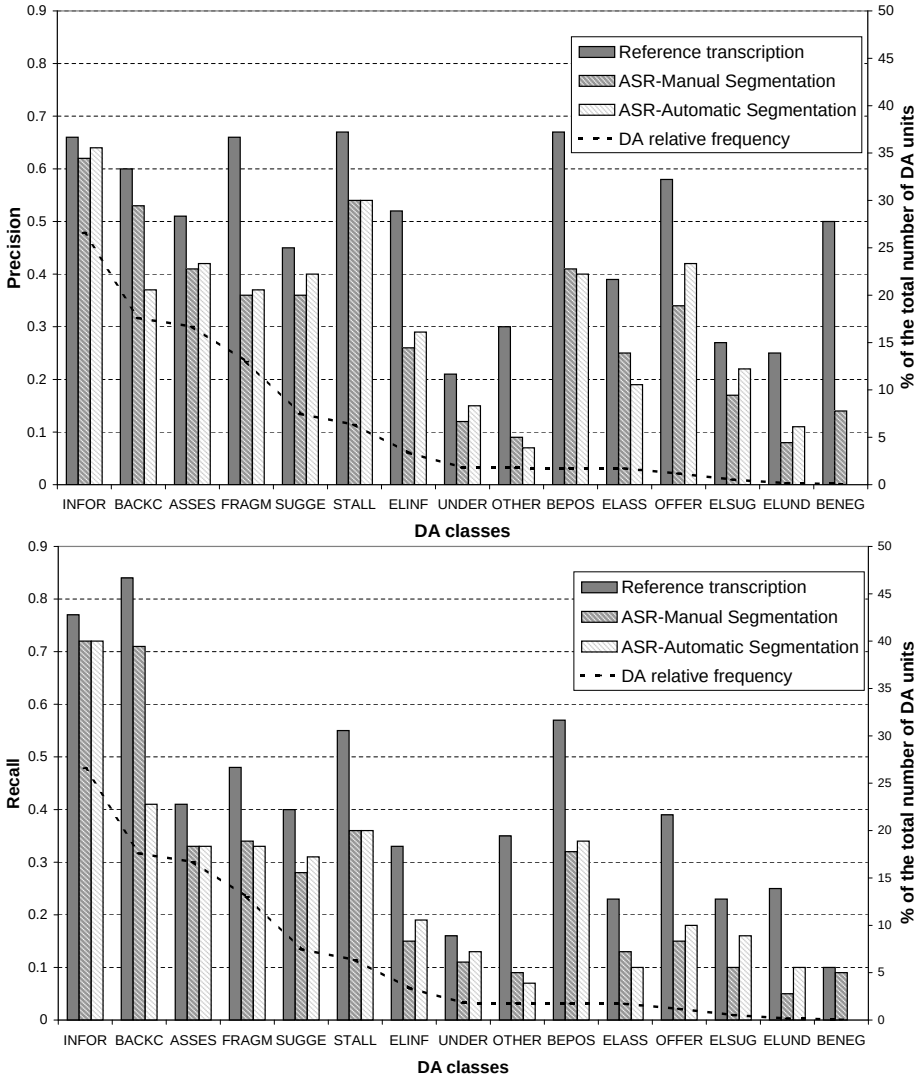


Figure 7.5: DA class based precision/recall metrics for the automatic DA tagging task on reference orthographic annotation and two versions of the ASR output. The 15 classes are sorted by their relative frequency in the AMI corpus (table 3.4), ranging from *inform* 26.6% (on the left) to *be-negative* 0.07% (on the right).

rate is slightly higher than the *FLM* approach, but on the joint recognition task, which involves both classification and segmentation, the *hybrid* provides the lowest errors.

Compared with the reference transcription, the automatically produced transcriptions, *ASR_AS* and *ASR_MS*, result in increased error rates for DA tagging, segmentation and recognition. For tagging, the *ASR_AS* system results in an increased CER of about 11% absolute, similar to that recorded on the ICSI tagging task (Dielmann and Renals, 2007c). Since the automatic DA segmentation strongly relies on the lexical content, a similar degradation can also be observed on DA segmentation metrics. The *iFLM* and *Hybrid* test conditions are less severely affected, suggesting that the larger language model results in a greater tolerance toward ASR inaccuracies. The full DA recognition task, representing a trade off between segmentation and classification, leads to an increase in the NIST-SU recognition metric by about 10% on *iFLM* and *Hybrid* setups and by 20% on the baseline *FLM* experiment.

However, the 12% of segments that are deleted in the *ASR_AS* transcription have an effect on the DA recognition results. In order to quantify this degradation, we compared the *ASR_AS* with the *ASR_MS* transcriptions which have an increased overall WER, but a reduced number of utterance deletions. Despite its higher WER, *ASR_MS* performs slightly better than *ASR_AS* on the isolated DA tagging task, although the lenient metric suggests that the situation is actually inverted when the DA classification is carried out as part of the joint DA recognition. Because of the lower number of deleted segments, *ASR_MS* outperforms *ASR_AS* on the DA segmentation sub-task using both the *FLM* and *iFLM* systems. A similar discourse applies to the overall recognition performances on the baseline *FLM* setup. Thanks to the more ASR tolerant interpolated FLM and to the improved *ASR_AS* transcription quality, which leads to better dynamic classification performances (Lenient metric), *ASR_AS* offers a slightly improved DA recognition over *ASR_MS* on both *iFLM* and *Hybrid* setups.

An example of the automatic DA recognition output (using the hybrid approach) is shown in figure 7.6. The reference manually annotated DA units (bold text) have been aligned to the automatic DA recogniser output produced using both the reference transcription (plain text) and the *ASR_AS* output (italic text). An excerpt rich

in interactions has been chosen for this example even if this often results in more ASR errors, because of overlapping speech and cross-talk between microphones, and thus in a lower DA recognition accuracy. The excerpt in figure 7.6 forms only an intuitive impression about the agreement between automatic DA recognition and manual DA annotation. More precisely, the agreement between hybrid FLM+iFLM DA recognition and reference DA annotation is given by: $k = 0.54$ (*REF*), $k = 0.45$ (*ASR_MS*), and $k = 0.48$ (*ASR_AS*). Note that the human inter-annotator agreement for the 15 DA AMI task is in the range $k = 0.83 - 0.89$ (section 3.2.3). DA recognition experiments on the AMI training set, reported in table 7.9, provide error rates proportional to those reported on the test set (table 7.7). In this case the agreement between hybrid approach and reference DA annotation is about $k = 0.79$ (*REF*), $k = 0.56$ (*ASR_MS*), and $k = 0.62$ (*ASR_AS*); a result still below the human inter-annotator agreement.

The switching DBN architecture generates both word sequences, using language models, and sequences of continuous prosodic features (using GMMs). We have performed a set of experiments to analyse the effect of the prosodic features. Table 7.8 gives tagging, segmentation and recognition results for the *manual*, *ASR_MS* and *ASR_AS* transcriptions, using a model that does not include the continuous prosodic features. The prosodic features do not contribute to the tagging task, hence the results in this case are unchanged. For the segmentation and recognition tasks it can be seen that removing the prosodic features results in a substantial increase in all the error rates, with the exception of the Lenient error metric.

7.8 Discriminative re-classification of joint recognition output

The use of high performances static discriminative classifiers to re-rank the output of sequential generative models has proven to be an effective technique in domains such as probabilistic parsing and statistical machine translation. In probabilistic parsing, a generative model estimates a list of parse hypotheses for each input sentence, then an additional discriminative model is used to rerank them (Collins, 2000; Collins and Koo, 2005; Koo and Collins, 2005). In statistical machine translation a

		Reference transcription			ASR manual segmentation			ASR automatic segmentation		
Task	Metric	FLM	iFLM	Hybrid	FLM	iFLM	Hybrid	FLM	iFLM	Hybrid
TAG.	100 - %Correct	40.9	51.4	42.8	50.7	61.2	53.0	52.7	61.9	54.8
S	NIST-SU	88.5	31.9	51.8	101.0	40.5	67.4	103.0	45.6	70.9
E	DSER	79.6	24.5	36.0	87.1	29.4	43.9	99.7	47.8	62.1
G	Strict	82.7	50.7	63.2	88.7	52.7	68.2	88.6	51.2	67.5
M.	Boundary	13.5	4.9	7.9	16.7	6.7	11.1	16.8	7.4	11.5
R	NIST-SU	109.2	85.4	102.0	120.0	99.7	124.4	120.6	99.2	123.4
E	DER	86.3	61.8	61.7	92.1	71.0	71.5	104.8	85.3	87.1
C.	Strict	88.0	74.8	77.1	93.0	79.3	82.9	92.9	78.4	82.3
	Lenient	40.6	51.4	44.0	44.1	57.7	50.6	43.0	55.9	49.7

Table 7.8: DA tagging, segmentation and recognition error rates (%) on the AMI meeting corpus without the use of continuous prosodic features; results are reported on 3 FLM setups both on reference and fully automatic ASR transcriptions.

Task	Metric	Reference transcription			ASR manual segmentation			ASR automatic segmentation		
		FLM	iFLM	Hybrid	FLM	iFLM	Hybrid	FLM	iFLM	Hybrid
S	NIST-SU	55.5	11.8	12.5	72.1	18.2	23.7	96.9	25.2	27.0
E	DSER	69.8	7.1	7.8	84.4	12.9	17.6	92.2	19.8	21.5
G	Strict	58.6	7.6	8.6	77.8	17.7	25.0	88.2	16.2	20.4
M.	Boundary	9.0	1.9	2.0	12.1	3.2	4.1	15.1	4.4	4.7
R	NIST-SU	78.1	33.9	33.2	93.6	63.7	66.9	111.6	63.3	63.4
E	DER	77.1	27.7	26.6	89.9	53.1	52.5	94.6	53.2	52.3
C.	Strict	66.6	14.8	16.0	84.6	43.6	47.5	91.2	38.8	41.5
	Lenient	32.6	11.9	12.4	43.8	33.5	33.1	42.5	29.3	29.1

Table 7.9: DA segmentation and recognition error rates (%) on the training set of the AMI meeting corpus; results are reported on 3 different FLM setups both on reference manual transcriptions and on 2 ASR outputs.

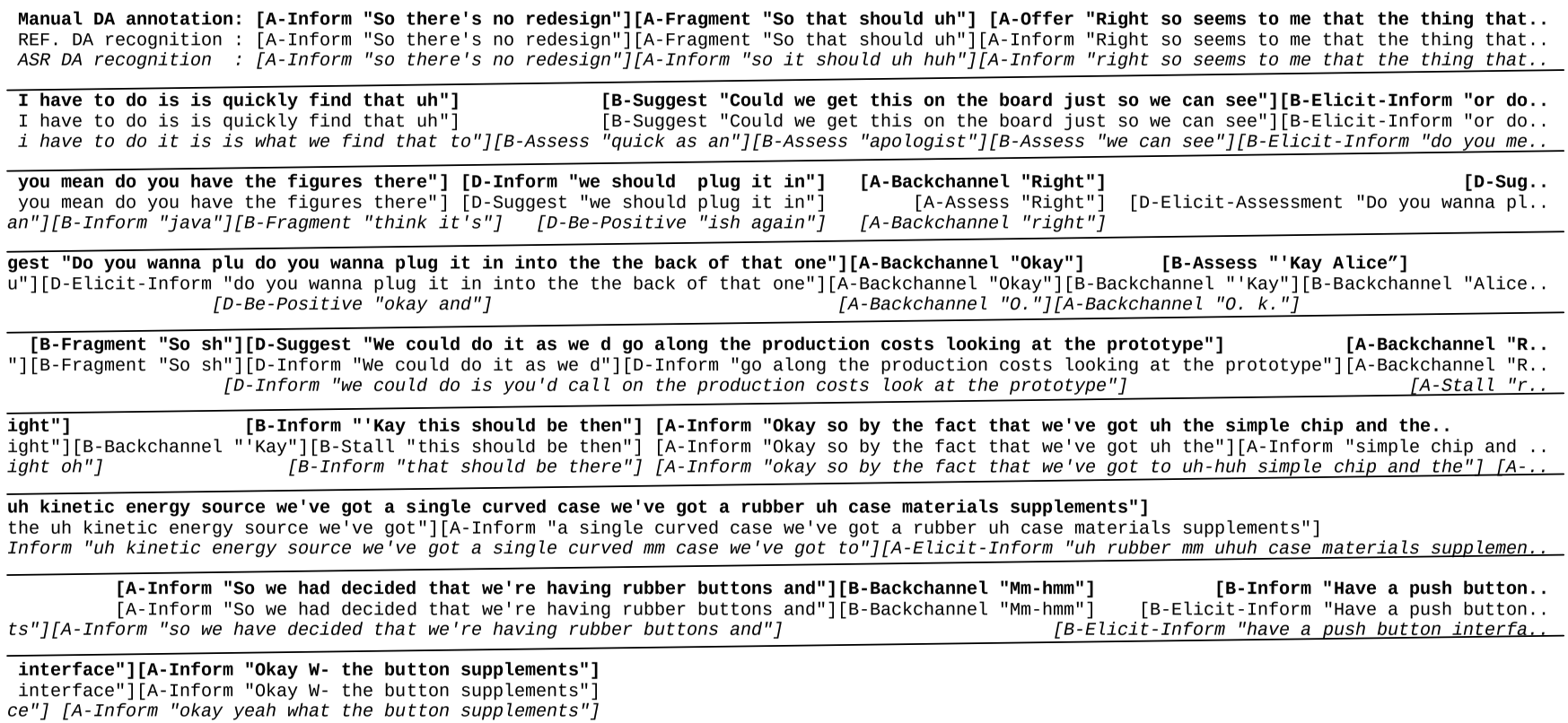


Figure 7.6: Manually annotated DA units from the AMI corpus (ES2014d) in bold (first row), and the automatic DA recogniser output obtained applying the switching DBN model with a *Hybrid* FLM configuration to the manual reference (second row) and the automatic ASR_AS transcriptions (third row, italic font). The DA segments have been specified using the following format: [Speaker label – DA label "utterance"] where the four interacting speakers have been represented through the capital letters A, B, C, and D.

similar approach could be used to rerank n-best lists of candidate translations (Shen et al., 2004).

Discriminative approaches have also been used to correct (or validate) the ASR transcription produced by a generative HMM system. Support Vector Machines trained on features related to the acoustics are used by Venkataramani et al. (2007) to disambiguate confusable word pairs. In another application of static reranking of LVCSR n-best hypotheses, additional phonetic, lexical, syntactic and semantic knowledge were used to discriminate between multiple recognition hypotheses (Balakrishna et al., 2006).

This is an attractive approach for several reasons. First, since it is a post-processing method it may be applied to any preexisting system leaving it unaltered. Second, directly discriminant approaches explicitly optimise an error rate criterion, while exploiting temporal boundaries and recognition candidates estimated by the generative model. Finally, it is possible to add features to the joint recognition system, with the possibility of lower computational overhead.

We have applied discriminative re-ranking to automatic DA recognition, post-processing the output of the *iFLM* system with a static discriminative classifier based on Conditional Random Fields (Lafferty et al., 2001). CRF are undirected graphical models globally conditioned on arbitrary long observation sequences. They are frequently used with a simplified *linear chain* topology (first-order CRF) which can be interpreted as a generalisation of HMMs. HMMs are generative models aimed at modelling the joint probability between observation and label sequences. Ideally the training of a generative model would require to enumerate every possible observation sequence, easily leading to an intractable problem. However in a HMM the observation at a given instant (i.e. the emission probabilities) is assumed to depend only on the current state. CRFs, aiming at modelling conditional probabilities over label sequences given a particular observation sequence, do not make specific (and often unwarranted) assumptions on the observations, allowing to model long-range dependencies of the observations and multiple interacting features. Moreover CRFs are trained discriminatively, i.e. each class is trained competing against all the other classes, and discriminative models for multiple classes are simultaneously trained. Since CRFs are discriminatively trained to maximise the conditional likelihood of a given training sequence and avoid unwarranted assump-

Recognition metrics	Reference transcription	ASR manual segmentation	ASR automatic segmentation
NIST-SU	59.3 (71.3)	72.6 (85.9)	71.8 (81.2)
DER	46.7 (51.9)	58.0 (62.5)	60.0 (64.1)
Strict	54.5 (62.1)	61.2 (68.5)	58.2 (64.7)
Lenient	36.5 (42.2)	43.2 (48.3)	41.7 (46.9)

Table 7.10: AMI DA recognition error rates (%) of a CRF based re-classification system without the use of discretised prosodic features. Best prior recognition performances using the *hybrid* approach have been reported in brackets.

Recognition metrics	Reference transcription	ASR manual segmentation	ASR automatic segmentation
NIST-SU	59.2 (71.3)	70.3 (85.9)	71.3 (81.2)
DER	46.7 (51.9)	56.1 (62.5)	59.7 (64.1)
Strict	54.2 (62.1)	59.3 (68.5)	57.4 (64.7)
Lenient	36.0 (42.2)	40.6 (48.3)	40.5 (46.9)

Table 7.11: AMI DA recognition error rates (%) of a CRF based re-classification system using lexical and prosodic features.

tions over the observations, they offer improved discrimination and a better support of correlated features. Moreover during CRF decoding the classification decision is taken globally over the entire sequence and not locally on a single observation.

The linear chain CRF has been used to associate DA labels with the best segmentations provided by the switching DBN. The prosodic features that we used in the generative model (with the exception of F0 variance) were discretised and used in conjunction with the lexical information during the CRF re-labeling process, implemented with CRF++ [7].

Tables 7.11 and 7.10 report the recognition performances [8] after discriminative re-classification, respectively with and without the adoption of discretised prosodic features. The improvement is consistent on all the transcription conditions and on

[7] Available from: `http://crfpp.sourceforge.net/`

[8] Discriminative CRF re-classification outputs showed a significant difference (at level $p = 0.001$ according to the MAPSSWE test) when compared to the input *iFLM* DA sequences.

all the evaluation metrics, with reduction of 5–12% absolute.

This improvement is mainly due to the discriminative use of the lexical content; the comparison between table 7.10 and 7.11 shows that prosodic features provide a marginal contribution of less than: 0.5% on reference transcriptions, 2.6% on *ASR_MS*, and 1.2% on *ASR_AS*. This confirms that acoustics related features can help to discriminate between DA units with similar lexical realisations, but word identities play a more central role in DA classification. The experiments reported in table 7.8 show that prosodic related features have a more substantial impact on the segmentation task, confirming the intuition behind exploiting the prosodic information in the switching DBN approach only for segmentation. This approximation also helped to reduce the model's complexity.

7.9 Discussion

We have presented a framework for the automatic recognition of dialogue acts in multiparty conversations. DA recognition experiments were carried out on the AMI meeting corpus using a dictionary of 15 DA classes tailored for decision making meetings, and on the ICSI corpus using a more generic set of 5 DA classes. The system that we have presented employs a generative probabilistic approach implemented through the integration of a heterogeneous set of technologies: six continuous prosodic features extracted from the lexical and prosodic content facilitate the segmentation process; a trigram discourse language model estimated from observed sequences of DAs; a factored language model interpolated using multiple conversational data resources, used in conjunction with a plain FLM trained solely on in-domain data; and a switching DBN model with two alternating topologies, which coordinates the joint DA segmentation and classification task by integrating the available resources. Multiple concurrent DA segmentation and classification hypotheses are evaluated by this joint DA recogniser, enabling the investigation of a larger search space compared with a two-step sequential segmentation-classification approach.

Three experimental systems were investigated based on a simple FLM, an interpolated FLM, or hybrid using both. The simple FLM trained only on data from the target AMI corpus offers the most accurate DA classification. However the in-

terpolated FLM, thanks to its richer dictionary and language model, reduces the number of segmentation errors by a factor of 2–3, at the cost of a slightly degraded DA classification accuracy. A hybrid approach, using both FLMs, allows a trade off between segmentation and classification, to improve the overall recognition accuracy. Experiments on AMI data using each of the three systems on hand-transcribed and two kinds of automatically transcribed data, showed that these systems generalise well to automatic imperfect transcriptions. A further significant improvement in the recognition accuracy, of 5–12%, was obtained using a discriminative DA re-classification process based on conditional random fields.

The degradation when moving from manual transcriptions to the output of a speech recogniser is less than 15% absolute for most tasks and metrics. Our experiments indicate that it is possible to perform automatic segmentation into DA units with a relatively low error rate, altough the system tends to over-segment (i.e. further subdvide the manually annotated reference segments, detecting a larger number of shorter DA units). However the operations of tagging and recognition into fifteen imbalanced DA categories have a relatively high error rate, even after discriminative reclassification, indicating that this remains a challenging task with a large potential for improvement (section 3.2.3). As the first complete set of joint DA recognition experiments reported on the AMI meetings, these experiments can also provide a baseline reference system for future work on this corpus.

Chapter 8

Improvements to Dialogue Act recognition

8.1 Introduction

The automatic system for the joint dialogue act recognition outlined in chapter 7 has successfully fulfilled its objectives achieving good recognition accuracies. However there is scope for improvement, and in this chapter we will present some enhancements to the switching DBN dialogue act recogniser.

Section 8.2 is concerned with the AMI joint DA recognition task, reporting on a novel set of experiments based on 4 broad DA categories obtained merging the original 15 DA classes. Moreover a procedure to improve DA classification by learning discriminative Factored Language Models will be proposed in section 8.3.

8.2 Further experiments on Dialogue Act recognition

Joint dialogue act segmentation and classification of the AMI meeting corpus was performed through an integrated framework based on a switching dynamic Bayesian network (section 7.6), discriminative conditional random fields based reclassification (section 7.8), and a set of continuous features and language models. The initial recognition experiments (sections 7.7.4 and 7.8) were based on a dictionary of 15 AMI DA classes tailored for group decision-making (section 3.2.3).

In section 8.2.1 we will outline some further DA recognition experiments us-

ing a reduced set of 4 broad DA classes. We will initially compare two approaches: training "from scratch" a new Switching DBN model using the reduced tag-set (section 8.2.2), and converting the output of the 15 DA Switching DBN system to 4 categories (section 8.2.3). In section 8.2.4, we will investigate the discriminative reclassification of both system outputs.

8.2.1 Joint Dialogue Act recognition using four broad DA categories

Dialogue Act annotation schemes often include a fairly large number of classes or rich hierarchical structures. The AMI annotation scheme (section 3.2.3) includes 15 specific DA classes; the ICSI MRDA scheme (section 3.2.2) and the Switchboard DAMSL scheme (section 3.2.4) constitute examples of the latter case. These large annotation schemes can be then reduced by merging together similar categories or by accounting only for the highest level of the DA hierarchy (section 3.2.2). The idea is to annotate the data with the richest possible scheme [1] and to reduce the number of classes according to the application domain. This allows an unlimited number of "virtual annotation schemes" to be dynamically built without the need of reannotating all the data. Note that type and number of DA classes required by each application (section 6.3) depend on the final application purpose, the overall accuracy of the DA recogniser, and the resulting joint performance of the fully automated system.

In this section we present some additional DA recognition experiments performed on the AMI corpus using a reduced number of DA categories. Early experiments of Hsueh and Moore (2007a,b) on automatic decision detection in conversational speech, suggested that replacing the 15 AMI DA classes with a reduced number of broader DA classes can improve decision detection. DA labels provide supporting evidence during the decision detection process, and are thus adopted as contextual features for a maximum entropy classifier. However not all the 15 labels play the same role on this task (Hsueh and Moore, 2007b): stall and fragment DAs tend to precede or follow decision making segments; elicit type DAs precede and follow non decision making sentences; decisions are more frequent within in-

[1] Although the trade-off between annotation costs and annotation accuracy needs to be carefully evaluated case by case.

Category	AMI DA classes	Proportion %
Category 1	*backchannel* (17.6%)	36.9
No speaker intention	*stall* (6.3%)	
	fragment (13.0%)	
Category 2	*inform* (26.6%)	50.8
	suggest (7.5%)	
	assess (16.7%)	
Category 3	*elicit inform* (3.4%)	5.8
Elicit classes	*elicit offer or suggestion* (0.5%)	
	elicit assessment (1.7%)	
	elicit comment understanding (0.2%)	
Category 4	*offer* (1.2%)	6.7
Other classes	*comment about understanding* (1.8%)	
	be positive (1.8%)	
	be negative (0.1%)	
	other (1.8%)	

Table 8.1: Four broad Dialogue Act categories obtained by merging the fifteen specialised AMI DA classes, with the percentage of DAs in each category.

form and suggest DAs. Therefore it is reasonable to cluster together the DA types which provide similar cues. Following these considerations, the original 15 AMI DA classes shown in table 3.4 can be grouped into a new set of 4 broad DA categories targeted on the automatic decision detection task. Table 8.1 shows the new 4 broad DA categories obtained by merging all DAs unrelated to specific speaker intentions (backchannel, stall, and fragment), by grouping information exchange DAs, forming a single class for elicit type DAs, and assigning all the remaining classes to a forth group.

The resulting 4 categories are unevenly distributed: information exchange accounts for more than half of the data, and elicit type DAs represent only 5.8% of the total number of DAs. Since the automatic mapping from 15 classes to 4 broad categories concerns only the DA labels but not their temporal segmentation, the original 15 DA manually annotated segmentation is preserved, thus both annotation schemes

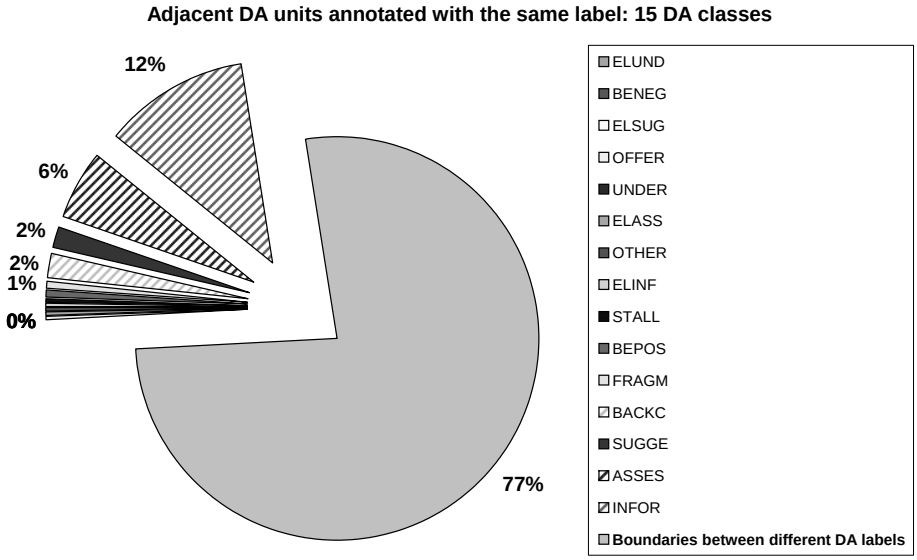


Figure 8.1: Dialogue act boundaries between units annotated with the same label: distribution using the 15 DA classes annotation scheme.

result in sharing the same segmentation. Note that adjacent segments are allowed to share the same DA label, observing for example two consecutive but independent "inform" DAs (15 classes case) or two consecutive elicit types (4 classes case). However reducing the number of classes from 15 to 4 increases the percentage of adjacent segments annotated with the same label. This is clearly evident comparing figures 8.1 and 8.2: using the original 15 classes annotation scheme only, 23.5% of the DA boundaries occur between DA units with the same label (figure 8.1); introducing the 4 categories scheme, more than 44% of the DA boundaries involve DA segments belonging to the same DA category (figure 8.2). Since "inform", "suggest", and "assess" frequently appear on both sides of a DA boundary, merging these 3 classes into the new "DA category 2" intensifies this phenomenon.

Three sets of automatic DA recognition experiments were performed on the AMI corpus using the new 4 broad DA categories. A new switching DBN system was trained from scratch and tested using the 4 DA reduced scheme (section 8.2.2). In a second experiment we converted the 15 classes recognition output obtained from the system of section 7.7.4 into 4 broad classes (section 8.2.2). Finally the best automatic DA segmentation is re-tagged using a Conditional Random Field classifier trained on 4 classes (section 8.2.4). Note that all experiments were performed both on orthographic reference transcriptions and on the fully automatic

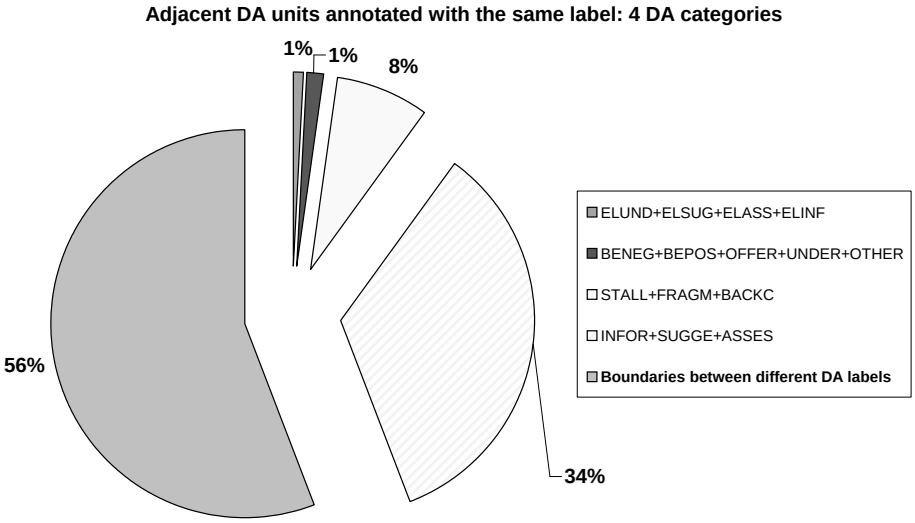


Figure 8.2: Dialogue act boundaries between units annotated with the same label: distribution using the 4 DA categories annotation scheme.

ASR output.

8.2.2 Switching DBN model trained on four broad DA categories

Similarly to section 7.7.4 we have used the switching DBN model for DA tagging, segmentation, and recognition adopting the same three language model configurations outlined in section 7.5.3: baseline FLM (*FLM*), interpolated FLM (*iFLM*), and a *hybrid* approach with the baseline and the interpolated FLM combined at decoding time. However the new system was trained using just four broad DA categories and tested both on reference and fully automatic orthographic transcriptions.

Error rates for the three tasks (DA tagging, segmentation, and recognition) using the three language model configurations and the two transcription conditions are reported in table 8.2 [2].

Comparing the 15 class system (table 7.7) to the 4 class approach (table 8.2) it is evident that the reduced number of DA classes results in a significant absolute improvement on the Classification Error Rate: about 18% on both transcription conditions and independently from the language model configuration. Class based precision and recall metrics for the 4 DA category tagging task are reported in figure

[2] MAPSSWE significance testing showed significant differences at level $p = 0.001$ between all the three systems.

		Reference transcription			ASR automatic segmentation		
Task	Metric	FLM	iFLM	Hybrid	FLM	iFLM	Hybrid
TAG.	100 - %Correct	**22.9**	31.5	24.8	**34.2**	42.7	36.9
S	NIST-SU	87.2	**19.6**	24.2	106.2	**27.7**	33.6
E	DSER	90.3	**14.5**	17.9	95.8	**24.3**	29.4
G	Strict	85.7	**29.4**	36.4	93.4	**27.8**	36.7
M.	Boundary	13.3	**3.0**	3.7	17.3	**4.5**	5.5
R	NIST-SU	97.3	52.7	**51.7**	112.1	61.2	**62.1**
E	DER	91.6	40.9	**38.8**	96.6	52.4	**51.9**
C.	Strict	87.1	48.2	**47.7**	94.1	50.1	**51.1**
	Lenient	17.6	27.3	**17.8**	17.2	31.3	**21.5**

Table 8.2: DA tagging, segmentation and recognition error rates (%) on the AMI meeting corpus using 4 broad DA categories; results are reported on 3 different FLM setups (baseline FLM, interpolated FLM, and hybrid FLM+iFLM) both on reference manual transcriptions and on fully automatic ASR transcriptions.

Task	Metric	Reference transcription			ASR automatic segmentation		
		FLM	iFLM	Hybrid	FLM	iFLM	Hybrid
S	NIST-SU	72.0	11.9	12.4	102.1	21.8	25.1
E	DSER	84.8	7.2	7.7	94.3	20.8	23.8
G	Strict	73.0	7.5	8.3	90.8	17.1	22.6
M.	Boundary	11.8	1.9	2.0	15.9	3.8	4.4
R	NIST-SU	83.7	27.3	27.9	109.5	47.1	49.1
E	DER	86.3	21.4	21.9	95.1	43.0	44.0
C.	Strict	74.7	12.2	13.3	91.6	30.6	34.3
	Lenient	17.4	7.3	7.8	18.8	17.1	15.5

Table 8.3: DA segmentation and recognition error rates (%) on the training set of the AMI meeting corpus using 4 broad DA categories; results are reported on 3 different FLM setups both on reference manual transcriptions and on fully automatic ASR transcriptions.

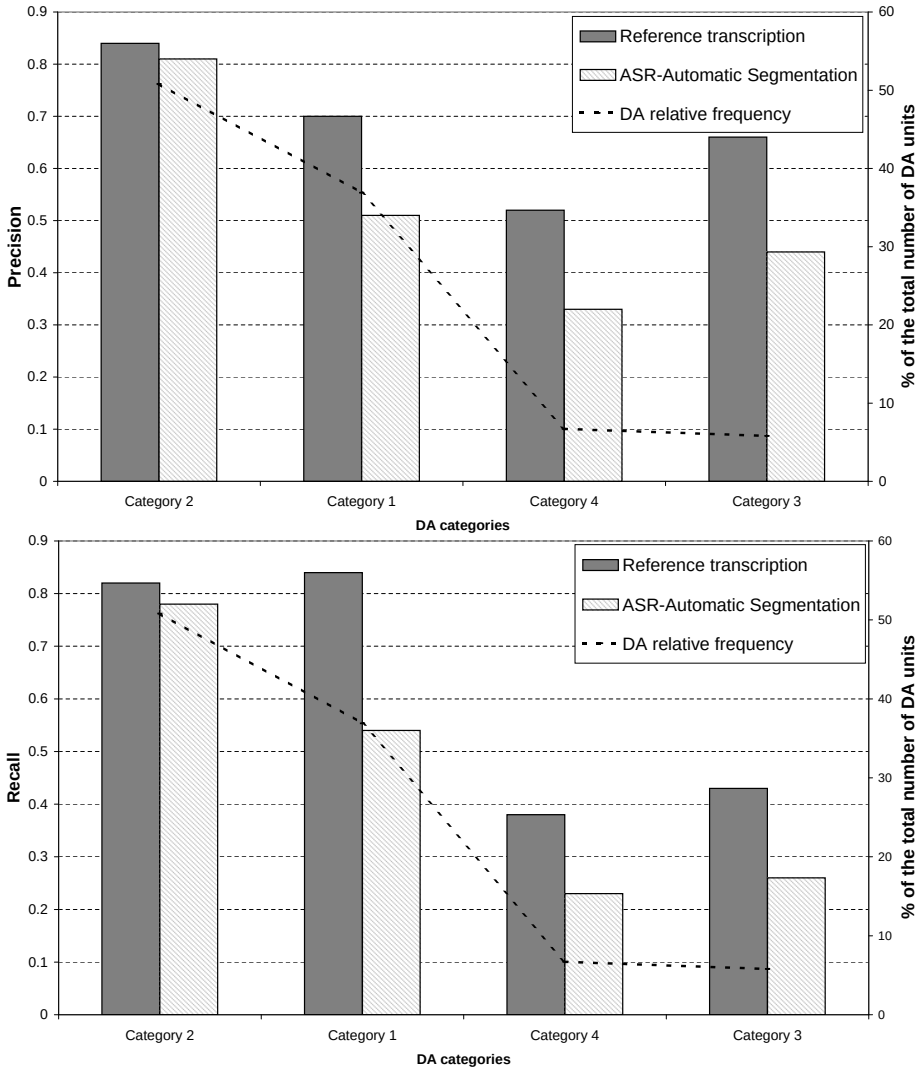


Figure 8.3: Precision/recall metrics for the automatic DA tagging task on reference orthographic annotation and ASR_AS output using 4 broad DA categories.

8.3. The classifier performs better on the two most frequent categories, showing the highest recall for *Category 1* and *Category 2*. This once again suggests that DA tagging is heavily influenced by the prior distribution of the classes. Not surprisingly precision and recall estimated using 4 broad DA categories are higher than those observed on 15 classes (figure 7.5).

Similarly to what has been previously observed using 15 classes, the baseline FLM offers better classification performances than the interpolated FLM and the hybrid approaches. However interpolating the FLM on additional data resources has a positive effect on all the segmentation metrics, and the hybrid system offers the best DA recognition performances even with 4 broad DA categories. This behaviour is also evident performing DA segmentation and recognition on the training data (table 8.3): interpolated FLM and hybrid approach scored the lowest error rates on the training set. Moreover a smaller DA dictionary resulted in an improved fit between statistical models and training data: this can be observed comparing the training set recognition error rates using 4 broad DA classes (table 8.3) and 15 DA labels (table 7.9).

Although the *iFLM* and the *hybrid* configurations achieved similar DA segmentation performances using 15 or 4 DA categories, the baseline *FLM* model seems to be more favorable toward 15 classes. The new segmentation obtained using a system trained on just 4 broad DA categories rather than 15 classes shows:

- 2% more false-alarms (DA boundaries wrongly inserted within a single DA unit);
- a decrease of about 14% in the number of correctly detected DA boundaries;
- 14% more missed boundaries (adjacent DA units which are erroneously merged together).

Using 15 classes 24.6% of the missed boundaries were associated to adjacent DAs with the same label, this percentage rises to 44.2% when the baseline *FLM* configuration is applied to 4 broad DA categories. This is related to the fact that the reference segmentation has been manually annotated using 15 classes, and the automatic mapping from 15 to 4 categories increases the number of adjacent DA units with the same labelling (section 8.2.1). These boundaries are more likely to be

missed, since there is less supporting evidence for a DA boundary, and the two consecutive segments of the same kind can be easily confused for a single longer DA unit. However the adoption of an interpolated FLM seems to address this issue effectively providing similar segmentation performances both using 15 and 4 DA categories.

This phenomenon is also evident on joint DA recognition performances: the highest recognition error rates are observed using the baseline *FLM* system; *iFLM* shows a consistent improvement over *FLM*; and the *hybrid* configuration sets the best trade-off between segmentation and classification.

8.2.3 Converting the 15 classes DA recognition output into 4 categories

The rule based mapping from 15 classes to 4 broad DA categories outlined in table 8.1 can also be applied to the automatically recognised DA units obtained using the 15 DA classes system discussed in section 7.7.4. Meetings are thus segmented and tagged using the extensive dictionary containing all the 15 symbols, then each recognised DA label is automatically mapped into one of the 4 broad DA categories.

Table 8.4 shows DA tagging, segmentation, and recognition error rates after this post-recognition conversion process. Note that the automatically detected DA boundaries are left unchanged during the conversion process, thus all the segmentation results scored using the original 15 DA system (table 7.7) are valid even after the conversion to 4 DA categories (table 8.4). The post-recognition conversion approach (table 8.4) achieves similar performances to the switching DBN model trained on 4 categories (table 8.2). However a close comparison leads to several observations:

- the post recognition mapping has a positive outcome on DA classification, with the exception of the baseline *FLM* system configuration;
- the automatic segmentation provided by the 15 DA *FLM* system is exempt from some conversion side effects, i.e. the increased number of adjacent segments with the same label (section 8.2.1);
- the improved segmentation granted by the *FLM* setup is also evident on the

		Reference transcription			ASR automatic segmentation		
Task	Metric	FLM	iFLM	Hybrid	FLM	iFLM	Hybrid
TAG.	100 - %Correct	**23.0**	23.8	16.9	**35.0**	33.3	28.4
S	NIST-SU	70.7	**20.4**	25.6	102.6	**30.7**	34.0
E	DSER	78.0	**12.8**	17.6	94.2	**23.2**	25.8
G	Strict	74.4	**28.5**	36.9	91.5	**26.9**	33.7
M.	Boundary	10.8	**3.1**	3.9	16.7	**5.0**	5.5
R	NIST-SU	83.8	52.5	**52.8**	109.6	62.7	**62.6**
E	DER	81.7	39.2	**37.3**	95.3	50.2	**49.0**
C.	Strict	77.3	44.7	**47.5**	92.5	46.0	**47.8**
	Lenient	16.5	23.0	**17.1**	17.5	26.2	**20.2**

Table 8.4: DA tagging, segmentation and recognition error rates (%) on the AMI meeting corpus using 4 broad DA categories; results obtained converting the output from the 15 classes DA recogniser of table 7.7 into 4 broad DA categories.

Recognition metrics	Reference transcription	ASR automatic segmentation
NIST-SU	46.6 (51.7)	57.7 (62.1)
DER	33.9 (38.8)	45.7 (51.9)
Strict	39.7 (47.7)	40.7 (51.1)
Lenient	15.5 (17.8)	17.9 (21.5)

Table 8.5: DA recognition error rates (%) of the CRF based re-classification applied to the 4 classes DA recogniser outlined in section 8.2.2. Best prior recognition performances using the *hybrid* approach of table 8.2 have been reported in brackets.

overall *FLM* recognition performances;

- on all 3 configurations (*FLM*, *iFLM*, and *hybrid*), a post-recognition conversion provides slightly improved performances than naïve 4 classes DA recognition.

Running a switching DBN system using 15 DA classes and converting its recognition output to 4 broad DA categories is practical and effective. It is also reasonable to expect that this behaviour is generalisable and can be observed on similar mappings from 15 classes to a reduced set of categories. Moreover the discriminative reclassification techniques proposed in section 7.8 can be adapted to 4 categories, further improving the overall recognition accuracy.

8.2.4 Re-classification using four broad DA categories

The use of a Conditional Random Field static discriminative classifier to re-estimate the output of a joint generative DA recogniser has been discussed in section 7.8, proving to be effective on the 15 AMI DA task.

This approach can be similarly applied to the output of the naïve 4 classes DA recogniser outlined in section 8.2.2. A linear chain CRF, trained on discretised prosodic features and on word identities, can be used to associate DA labels drawn from the dictionary of 4 broad DA categories to the best segmentation output provided by the switching DBN of section 8.2.2. Table 8.5 reports the recognition performances using 4 broad categories after discriminative re-classification. The

Recognition metrics	Reference transcription	ASR automatic segmentation
NIST-SU	42.5 (52.8)	53.9 (62.6)
DER	33.2 (37.3)	44.8 (49.0)
Strict	39.3 (47.5)	39.8 (47.8)
Lenient	13.1 (17.1)	15.6 (20.2)

Table 8.6: DA recognition error rates (%) of the CRF based re-classification applied to the 4 classes post-recognition conversion outlined in section 8.2.3. Best prior recognition performances have been reported in brackets.

best segmentation obtained using the *iFLM* setup in table 8.2 has been re-classified using a linear CRF trained on 4 DA categories; a consistent improvement [3] can be observed on all the evaluation metrics, yielding an absolute reduction in the range of 2–10%.

The same 4 DA categories re-classification process can be applied to the best 15 DA segmentation obtained using the switching DBN recogniser of section 7.7.4 (*iFLM* configuration of tables 7.7 and 8.4). Table 8.6 shows the recognition performances achieved following this procedure. Similarly to the previous re-classification experiment, discriminative re-classification resulted in an absolute reduction of 4–10% according to the recognition metric.

DA segmentation using a switching DBN targeted on 15 classes, followed by CRF based re-classification using just 4 categories, provides the best performances on the AMI 4 broad DA recognition task.

8.3 Discriminative Factored Language Models

FLMs were adopted by the switching DBN DA recognition system outlined in chapter 7 to relate word identities and DA labels, thus learning the relationships between sentences and their enclosing dialogue act labels. The resulting FLMs are directly responsible for the tagging accuracy of the whole DA recogniser, and their ability to discriminate between different DA labels is the main objective function that needs

[3] Similarly to section 7.8, baseline generative DA recognition and discriminative CRF re-classification showed significant differences at level $p = 0.001$ according to the MAPSSWE test.

to be optimised (section 7.5.2).

Being interested in improving the overall DA classification accuracy, a procedure to construct more discriminative FLMs has been investigated. This method stems from the conventional FLM training procedure adopted in chapter 7. The training material is preventively processed estimating n-gram counts for each bundle of factors defined by the FLM topology. Then, instead of directly building the language model from these estimates, the n-gram counts are rescored favoring those which lead to positive DA classifications and penalising all the n-gram counts responsible for DA tagging errors. The aim of this rescoring step is to improve the discrimination between different DA labels, thus developing novel Discriminative Factored Language Models. The discriminative FLM training is implemented through an iterative procedure based on 3 steps:

1. DA classification hypotheses generation: the training dataset is classified in term of DAs using the FLM learned during the previous iteration cycle and the SRILM based DA classifier outlined in section 7.5.2;

2. n-gram counts rescoring: counts responsible for correct DA predictions are enhanced and wrong classifications penalised;

3. FLM generation: a FLM is built from the rescored counts.

A conventionally trained FLM is used to bootstrap this iterative method, and the recursive process is stopped when the DA classification error rate estimated on the development set converges to a stable value (no significant improvements are observed).

The experimental DA classification results, achieved using this discriminative approach, are reported in table 8.7. The first two rows show the classification error rate on the 5 broad DA ICSI task, both using reference and automatic orthographic transcriptions. The highest improvement over the baseline model is observed after a single iteration, and the discriminative FLM is responsible for an absolute improvement between 0.7% and 1.3%. Similarly the last two rows investigate discriminative FLM training in the context of the AMI 15 DA tagging task. An absolute improvement of 1.3% is observed on the reference transcription condition, after two discriminative training iterations. The ASR test condition benefits from a smaller

			Discriminative Factored Language Model			
Corpus	Transcription	FLM	1^{st} iteration	2^{nd} iteration	3^{rd} iteration	4^{th} iteration
ICSI	Reference	29.1	**28.4**	28.5	28.4	28.5
ICSI	ASR	38.1	**36.8**	36.8	36.8	36.8
AMI	Reference	47.7	46.5	**46.4**	46.4	46.4
AMI	ASR_AS	59.7	**58.7**	58.7	58.7	58.7

Table 8.7: DA tagging error rate (%) on the ICSI and AMI meeting corpora using the Discriminative Factored Language Model alone. Results are reported after the first 3 iterations of discriminative training (both using reference and automatic transcriptions) and compared to the baseline FLM.

improvement of 1% absolute. Note that all the DA classification results reported on table 8.7 were obtained employing the same experimental setup introduced in section 7.5.2.

Discriminative FLMs are thus responsible for an absolute improvement of about 1% in terms of DA classification error rate. Moreover this improvement is usually observed after less than 3 iteration cycles of discriminative training.

8.4 Discussion

In this chapter we have outlined some experimental dialogue act recognition results using a reduced AMI DA annotation scheme based on 4 broad categories instead of 15 DA classes. The switching DBN DA recogniser, presented in chapter 7, was effective even on this novel task, further validating this approach which had already scored positive results both on the ICSI 5 broad DA task (section 7.7.3) and on the original AMI 15 DA task (section 7.7.4). Moreover in section 8.3 an improved training procedure for the factored language models has been outlined. The resulting Discriminative FLMs improved the DA classification accuracy by about 1% absolute both on automatic and reference orthographic transcriptions.

There is space for further enhancements and the dialogue act recogniser can be improved in order to operate in real-time under strict latency constraints, facilitating automatic online meeting structuring. In section 6.3 we presented several applications which can benefit from dialogue act recognition, such as topic detection and tracking, summarisation, decision detection, and automatic speech recognition. The adoption of application-specific DA related features, like the 4 broad DA categories targeted on topic detection presented in section 8.2.1, can be investigated in other relevant domains such as speaker addressing (Jovanovic et al., 2006).

Chapter 9

Conclusions

Multi-party meetings are a natural form of interaction in which different subjects share their thoughts, decisions, and ideas following an agenda and trying to fulfill a set of concurrent tasks. Meetings are sociological events, in which a large amount of information is generated and shared between a group of participants. Therefore an automated system to capture, store, structure and index meetings, is useful to:

- spread knowledge between people who have missed the meeting
- preserve meeting contents, avoiding confusions and omissions, enabling meeting participants to recall details
- facilitate remote meeting participation
- understand the structure of meetings in terms of temporal evolution, decision taking process, and topic structure.

We can simply capture meeting contents, through multi-perspective and multi-channel audio-video recordings. However, without further analysis, the semantic content of the meeting remains locked in an intractable low-level multimodal data stream. Orthographic transcription of speech in meetings represents a further step in this task. Meetings are a case of spontaneous human interaction, and their transcriptions tend to be redundant and only partially able to highlight the underlying meeting structure. The automatic structuring of meetings is a complex task that intersects many research areas, including automatic speech recognition, gesture recognition, topic segmentation, and emotion detection. Our goal was to develop a general purpose

framework to structure meetings detecting both individual intentions and group social interactions.

Meeting participants not only show individual behaviours, but also take part in a more general group behaviour. We were both interested in analysing such collective behaviour and in studying how conversations progress across time. Our aim was to develop an automatic approach to highlight when the group is discussing a topic or taking some notes, or when an individual meeting participant is reporting to the group, facilitating his presentation by using a white-board, or showing some slides. We also investigated the automatic structuring of a conversation, developing an infrastructure to recognise the atomic building blocks of a dialogue such as questions, statements, acknowledgments, and offers.

9.1 Summary

Two similar tasks were addressed in this thesis: automatic meeting segmentation using a dictionary of five group meeting actions, and automatic structuring of multiparty conversations in terms of dialogue acts.

9.1.1 Group meeting action recognition

The first task is concerned with the automatic segmentation of meetings into a sequence of meeting actions or phases, such as monologue, dialogue, note-taking, presentation and presentation at the whiteboard. We investigated the automatic recognition of meeting actions which involve the whole group and are independent from who is attending the meeting. Thus we need to identify the set of clues in both individual and group behaviours, and to highlight repetitive patterns in the communicative process. These may then be integrated into the abstract concept of meeting actions. We adopted a statistical approach based on four feature families and a Bayesian network infrastructure. Three multimodal feature vectors related to prosody and lexical content, speech, and visual activity were extracted from the raw audio-video recordings. Denoised estimates of F0, syllabic rate of speech, speech signal energy, and the output of a monologue/dialogue discriminator constituted the first feature vector. The second stream of features included a combination of 6 loca-

tion based speech activities over the past 3 frames, aiming at capturing the speaker turn taking dynamic. Head and hand average motion intensity and direction formed the third data stream.

Meeting action recognition experiments were performed on the M4 corpus (section 3.2.1), a collection of short multiparty meetings, consisting of more than five hours of audio-video recordings. Preliminary experiments using a baseline hidden Markov model showed that speaker turn features provide the highest percentage of correctly recognised actions, followed by lexical, prosodic, and visual features. Early combination of the 4 feature families into a single observation vector also proved to be effective, suggesting a complementarity between different modalities.

The 4 feature families were reduced to 3 feature streams by combining prosodic and lexical features. Then these 3 observation sequences were separately modelled through a multistream DBN approach. In this hierarchical infrastructure each feature stream is processed independently by the lowest layer of the model, and the partial information is integrated by the upper stage of the model. Compared to a baseline HMM, this technique models each feature stream individually, providing an improved control over the state-space, and allowing the model to encompass complex interdependences between different modalities. This approach also avoids an early integration of the 3 observation streams, delaying the information integration point to the last stage of the processing. Investigations using mixed integration points were also conducted (section 5.4.5). We also explored the use of a counter structure, this extension to the model aims to explicitly model state durations in order to constrain the number of action transitions.

The proposed multistream DBN architecture showed a significant improvement over a baseline HMM, with the multistream approach attaining an accuracy of 89.1% and producing an error rate of 12.2%. The multistream architecture was also validated on 3 independent feature setups (section 5.5): a subset of our feature collection, and two independent feature sets provided by IDIAP and Technische Universität München research institutes. The proposed DBN model achieved good recognition accuracies on all the 3 feature setups, proving its flexibility toward different feature sets, and suggesting that this is a principled approach to integrate multiple feature streams.

The output of the meeting action recogniser can be used to facilitate information

extraction tasks such as topic detection and tracking, automatic summarisation, or even to improve automatic speech recognition by explicitly modelling the current communicative context. Applications such as audio-visual summarisation–the editing of raw meeting recordings to create a nice looking meeting summary–can also be investigated. Moreover the proposed set of 5 group meeting actions is only one of the possible dictionaries which could be used to highlight the meeting structure. McGrath (1991) proposed to codify meetings using a dictionary of 12 high-level symbols resulting from the combination between four modes of activity (inception, problem solving, conflict resolution and execution) and three functions (production, well-being and member support). This is a highly refined general-purpose categorisation which could provide further valuable insights on the meeting structure.

9.1.2 Dialogue act recognition

Our second task consisted in developing a framework for the automatic recognition of dialogue acts in multiparty conversations. Dialogue acts represent the function that utterances serve in a conversation and aim to capture the intentions of a single speaker. A DA annotation scheme provides a set of disjoint classes that may be used to label every possible conversational act. Our DA recognition experiments concentrated on three tasks: recognition of 5 broad DA categories on the ICSI meeting corpus (section 3.2.2), 15 DA classes tailored to the AMI scenario meetings (section 3.2.3), and 4 broad DA categories obtained from the original 15 AMI DA classes (section 8.2.1).

We focused on the joint DA recognition task, developing a system able to perform DA segmentation and classification in parallel. Our approach is based on the integration of a heterogeneous set of resources through a specialised switching DBN infrastructure. Similarly to the meeting action recognition task, a set of prosody related features, such as pitch, energy, word length, pause duration, and word informativeness, were extracted from the audio recordings. These features aimed at facilitating the DA segmentation process. An initial set of experiments based on the ICSI corpus (section 7.7.3) showed the impact of prosodic features on DA recognition, highlighting the effectiveness of pauses for a correct DA segmentation. Further experiments, conducted on the 15 DA AMI task, removing the prosodic observations, confirmed the importance of these continuous features.

A trigram discourse language model, trained on manually annotated DA units from the training dataset, was adopted to estimate the probability of a given sequence of DA labels. A factored language model, based on word identities, word position, and DA labels, was adopted to implement the mapping from DA labels to word sequences. The 3 factors employed by the FLM were chosen after some preliminary DA classification experiments using the FLM alone (section 7.5.2). The principal function of the FLM is to perform DA classification. However, thanks to the switching DBN framework, DA segmentation and classification are jointly optimised: our system selects the most likely sequence of labelled DA units among multiple segmentation hypotheses. In order to train the FLM on a larger set of examples, obtaining a richer vocabulary and improved n-gram counts, we investigated the interpolation of multiple FLMs trained on additional conversational data resources. For example the ICSI and Fisher corpora (section 3.2.4) were adopted to enrich the FLM used to classify AMI DA units. Since the DA annotation in terms of the 15 AMI DA classes is currently not available for the ICSI and Fisher corpora, these additional data resources were artificially annotated labeling every sentence with all the 15 possible DA labels in the AMI DA annotation scheme. This procedure allowed to exploit large unannotated data resources to extent the original FLM, and can be extended to similar text segmentation tasks.

Continuous features, discourse model, plain FLM, and interpolated FLM, were integrated through a modular switching DBN infrastructure. This statistical model coordinates the joint DA recognition task, evaluating multiple segmentation and classification hypotheses. Therefore a joint approach allows to explore a larger search space if compared to a sequential system, where a single segmentation hypothesis is evaluated.

Numerical experiments were performed both on the ICSI and AMI meetings using three FLM configurations: a plain FLM trained on in-domain data, a weighted interpolated FLM trained on additional conversational resources, and a hybrid setup combining both FLMs during decoding. The plain FLM offered the most accurate DA classification. The interpolated FLM resulted in a slightly reduced tagging accuracy and a considerable improvement in segmentation accuracy, with the number of DA segmentation errors being halved. Since the baseline FLM offered a good tagging error rate and the interpolated FLM provided an excellent segmentation,

their combination helped integrating these complementary strengths. The resulting hybrid configuration had average tagging and segmentation performances, but also provided the best DA recognition output. These behaviours were observed on all the three DA recognition tasks (5 ICSI DA categories, 15 AMI DA classes, and 4 AMI DA categories) both using reference and automatic transcriptions. Results on automatically transcribed data showed that the proposed architecture generalises well to imperfect transcriptions, with less than 15% of degradation for most task and metrics. The switching DBN framework proved to be a principled and effective approach to integrate multiple language models and data streams. Recognition performances on the 5 DA ICSI task suggested that our switching DBN approach constitutes a competitive framework for the joint DA recognition, performing well in comparison with the state of the art (Zimmermann et al., 2006b). Discriminative reclassification of the best segmentation hypotheses, using a conditional random field DA classifier, resulted in a further absolute reduction of 4-12% on the DA recognition error rates. However automatic DA recognition, with its strict evaluation procedures and relatively high error rates, proved to be a challenging task even after discriminative reclassification.

Dialogue act segments have been successfully employed in automatic summarisation (Murray and Renals, 2006), action item detection (Purver et al., 2007), decision detection (Hsueh and Moore, 2007b), automatic speech recognition (Jurafsky et al., 1997a; Taylor et al., 1998), and machine translation (Lee et al., 1997; Levin et al., 2003).

9.2 Conclusions

Group meeting action and dialogue act recognition represent two different granularities of a similar task: low-level (DAs), and an abstract (meeting actions) representation of the same communicative process. We hypothesised that a similar methodology can be successfully applied to both tasks, allowing to share features, model structures, and evaluation procedures.

A common set of features, aiming at highlighting the prosodic structure and the lexical content of the conversation, was employed on both tasks. These core features were used in conjunction with domain specific cues, such as the “speaker turn

features", which proved to be highly effective on the group action recognition task (section 5.4.3). For dialogue act recognition, the availability of time labeled orthographic transcriptions allowed the inclusion of features related to word length and pause duration, which proved to be advantageous for the automatic DA segmentation (section 7.7.3). The investigation of visual related features was limited to the meeting action recognition task. Compared to the audio related features, head/hand motion features played a marginal role (section 5.4), highlighting the predominance of speech in meetings. However visual cues may play a leading role on other meeting processing related tasks, for example head pose estimation outperforms voice activity detection on focus of attention tracking (Stiefelhagen, 2002). All the features proposed in this thesis, although focused on multiparty human interaction modelling, are applicable to other speech and image processing related tasks.

Both recognition tasks needed to integrate a multitude of knowledge sources, such as multiple loosely synchronised multimodal feature streams (section 3.4), joint probabilities estimated through specialised language models (section 7.5), and hard-coded deterministic rules (section 7.6). A generative probabilistic framework based on dynamic Bayesian networks offered a flexible approach to integrate these technologies. Graphical models offer an extensible and scalable methodology to develop elaborate statistical models and to implement complex architectures. Their internal topology, encoding conditional independence assumptions between variables, captures some knowledge about the problem including it within the model. Compared to conventional hidden Markov models, DBNs allow to efficiently factorise the state-space over a set of hidden random variables, and to subdivide the observation space into multiple feature streams. Each data stream can be processed independently and their knowledge integrated at the desired level of the model, as demonstrated by the two extended multistream DBN models (section 5.4.5). However this wide flexibility is paid in terms of high computational costs and large memory requirements, resulting in a limited scope of application for these graphical approaches. As outlined in section 2.4.5 the adoption of a large state-space HMM for complex tasks (such as large vocabulary continuous speech recognition) is often preferred to a compact but computationally expensive DBN.

Combining multiple features streams from different modalities proved to be very useful both for group meeting action recognition and joint dialogue act segmen-

tation and classification. On the first task the proposed multistream DBN approach allowed to concurrently process multiple interdependent feature streams. Its hierarchical structure seamlessly combined independent low-level feature processing and late information integration: each feature stream is governed by a private set of "subactions", these form the bases for the group meeting actions which are recognised by the highest level of the model. Similarly, the switching DBN model adopted for DA recognition played a pivotal role in integrating multiple knowledge sources. Prosodic features, being crucial for the DA segmentation task (section 7.7) and consequently for the entire recognition process, were modelled through a Gaussian mixture model conditioned on the DA boundary detection subsystem. Knowledge from the orthographic transcription is included through a factored language model, which implements the mapping between word sequences and DA labels. Finally a discourse language model (section 7.4) represents the sequence of DA labels. Prosodic cues and LM probabilities are exploited by the DBN infrastructure to evaluate multiple segmentation and labelling hypotheses. Therefore the switching model not only integrates multiple data sources but also actively coordinates the whole DA recognition process.

Dynamic Bayesian Networks allowed to develop rich statistical models with a complex state-space, proving to be effective at integrating multiple feature streams related to different communicative modalities. The features and the models developed in this thesis, although focused on studying multiparty human-human interactions, are generalisable and translate well to other research domains. In particular the concept of integrating multiple resources, such as multiple feature streams and language models, through a DBN framework can be investigated on numerous applications.

Bibliography

Al-Hames, M., Dielmann, A., Gatica-Perez, D., Reiter, S., Renals, S., Rigoll, G., and Zhang, D. (2006a). Multimodal integration for meeting group action segmentation and recognition. In *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI–05)*, pages 52–63. Springer.

Al-Hames, M., Hörnler, B., Müller, R., Schenk, J., and Rigoll, G. (2007a). Automatic multi-modal meeting camera selection for video-conferences and meeting browsers. In *Proc. IEEE ICME*, pages 2074–2077.

Al-Hames, M., Hörnler, B., Scheuermann, C., and Rigoll, G. (2006b). Using audio, visual, and lexical features in a multi-modal virtual meeting director. In *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI–06)*, pages 63–74. Springer.

Al-Hames, M., Lenz, C., Reiter, S., Schenk, J., Wallhoff, F., and Rigoll, G. (2007b). Robust multi-modal group action recognition in meetings from disturbed videos with the asynchronous hidden Markov model. In *Proc. IEEE ICIP*, volume 2, pages 213–216.

Al-Hames, M. and Rigoll, G. (2005a). A multi-modal graphical model for robust recognition of group actions in meetings from disturbed videos. In *Proc. IEEE ICIP*, volume 3, pages 421–425.

Al-Hames, M. and Rigoll, G. (2005b). A multi-modal mixed-state dynamic Bayesian network for robust meeting event recognition from disturbed data. In *Proc. IEEE ICME*, pages 45–48.

AMI DA Annotation Guidelines (2005). Guidelines for Dialogue Act and Ad-

dressee Annotation V.1.0. Available from: `http://mmm.idiap.ch/private/ami/annotation/dialogue_acts_manual_1.0.pdf`.

Ang, J., Liu, Y., and Shriberg, E. (2005). Automatic dialog act segmentation and classification in multiparty meetings. In *Proc. IEEE ICASSP*, volume 11, pages 1061–1064.

Arnborg, S., Corneil, D. G., and Proskurowski, A. (1987). Complexity of finding embeddings in a k-tree. *SIAM Journal on Algebraic and Discrete Methods*, 8:277–284.

Balakrishna, M., Moldovan, D., and Cave, E. (2006). N-best list reranking using higher level phonetic, lexical, syntactic and semantic knowledge sources. In *Proc. IEEE ICASSP*, volume 1, pages 413–416.

Baron, D., Shriberg, E., and Stolcke, A. (2002). Automatic punctuation and disfluency detection in multi-party meetings using prosodic and lexical cues. In *Proc. ICSLP*, Denver, Colorado, USA.

Basu, S., Choudhury, T., Clarkson, B., and Pentland, A. (2001). Towards measuring human interactions in conversational settings. In *Proc. HLT–NAACL*.

Baum, L. (1972). An inequality and associated maximization technique in statistical estimation of probabilistic functions of Markov processes. *Inequalities*.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41(1):164–171.

Beeferman, D., Berger, A., and Lafferty, J. (1999). Statistical models for text segmentation. *Machine Learning*, 31:177–210.

Bengio, S. (2003). An asynchronous Hidden Markov Model for audio-visual speech recognition. *Advances in Neural Information Processing Systems (NIPS)*, 15.

Bengio, Y. and Frasconi, P. (1995). An input output HMM architecture. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, volume 7, pages 427–434.

Bhagat, S., Carvey, H., and Shriberg, E. (2003). Automatically generated prosodic cues to lexically ambiguous dialog acts in multiparty meetings. In *Proc. International Congress of Phonetic Sciences*, pages 2961–2964.

Bilmes, J. (1999). Buried Markov models for speech recognition. In *Proc. IEEE ICASSP*, volume 2, pages 713–716.

Bilmes, J. (2000). Dynamic Bayesian multinets. In *Proc. International Conference on Uncertainty in Artificial Intelligence*.

Bilmes, J. (2003). Graphical models and automatic speech recognition. *Mathematical Foundations of Speech and Language Processing*.

Bilmes, J. and Bartels, C. (2005). Graphical model architectures for speech recognition. *IEEE Signal Processing Magazine*, pages 89–100.

Bilmes, J. and Kirchhoff, K. (2003). Factored language models and generalized parallel backoff. In *Proc. HLT–NAACL*, volume 2, pages 4–6.

Bilmes, J. and Zweig, G. (2002). The Graphical Model ToolKit: an open source software system for speech and time-series processing. In *Proc. IEEE ICASSP*, volume 4, pages 3916–3919.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

Bolt, R. A. (1980). "Put-that-there": Voice and gesture at the graphics interface. In *Proc. conf. on Computer graphics and interactive techniques (SIGGRAPH)*, pages 262–270.

Bourlard, H. A. and Morgan, N. (1993). *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Academic Publishers.

Boyen, X. and Koller, D. (1998). Tractable inference for complex stochastic processes. In *Proc. International Conference on Uncertainty in Artificial Intelligence*, pages 33–42.

Brand, M., Oliver, N., and Pentland, A. (1997). Coupled hidden Markov models for complex action recognition. In *Proc. IEEE CVPR*, pages 994–999.

Carletta, J. (1996). Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., McCowan, I., Post, W., Reidsma, D., and Wellner, P. (2006). The AMI meeting corpus: A pre-announcement. In *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI–05)*, pages 28–39. Springer.

Carletta, J., Isard, A., Isard, S., Kowtko, J. C., Doherty-Sneddon, G., and Anderson, A. H. (1997). The reliability of a dialog structure coding scheme. *Computational Linguistics*, 23:13–31.

Çetin, Ö. (2004). *Multi-rate Modeling, Model Inference, and Estimation for Statistical Classifiers*. PhD thesis, University of Washington.

Cieri, C., Miller, D., and Walker, K. (2004). The Fisher corpus: a resource for the next generations of speech-to-text. In *Proc. LREC*, pages 69–71.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Collins, M. (2000). Discriminative reranking for natural language parsing. In *Proc. International Conf. on Machine Learning*, pages 175–182.

Collins, M. and Koo, T. (2005). Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., and Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems*. Springer.

Dai, P., Di, H., Dong, L., Tao, L., and Xu, G. (2007). Group interaction analysis in dynamic context. *IEEE transactions on Systems, Man, and Cybernetics*, 38(1):275–282.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

Dhillon, R., Bhagat, S., Carvey, H., and Shriberg, E. (2004). Meeting recorder project: Dialogue act labeling guide. Technical Report TR-04-002, International Computer Science Institute (ICSI), Berkeley, CA.

Dielmann, A. and Renals, S. (2004a). Dynamic Bayesian networks for meeting structuring. In *Proc. IEEE ICASSP*, pages 629–632.

Dielmann, A. and Renals, S. (2004b). Multi-stream segmentation of meetings. In *Proc. IEEE Workshop on Multimedia Signal Processing*, pages 167–170.

Dielmann, A. and Renals, S. (2007a). Automatic meeting segmentation using dynamic Bayesian networks. *IEEE Transactions on Multimedia*, 9(1):25–36.

Dielmann, A. and Renals, S. (2007b). DBN based joint dialogue act recognition of multiparty meetings. In *Proc. IEEE ICASSP*, volume 4, pages 133–136.

Dielmann, A. and Renals, S. (2007c). Multistream recognition of dialogue acts in meetings. In *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI–06)*, pages 178–189. Springer.

Duh, K. (2005). Jointly labeling multiple sequences: A factorial HMM approach. In *Proc. ACL 2005, Student Research Workshop*.

Duh, K. and Kirchhoff, K. (2004). Automatic learning of language model structure. In *Proc. International Conference on Computational Linguistics (COLING)*. Article 148.

Dupont, S. and Luettin, J. (2000). Audio-visual speech modeling for continuous speech recognition. *IEEE Transactions on Multimedia*, 2(3).

Ferguson, J. D. (1980). Variable duration models for speech. In *Proc. of the Symposium on the Applications of Hidden Markov Models to Text and Speech, IDA-CRD*, pages 143–179, Princeton, NJ.

Fernandez, R. and Picard, R. (2002). Dialog act classification from prosodic features using support vector machines. In *Proceedings of speech prosody 2002*.

Fine, S., Singer, Y., and Tishby, N. (1998). The hierarchical hidden Markov model: analysis and applications. *Machine Learning*, 32(1):41–62.

Galley, M., McKeown, K. R., Fosler-Lussier, E., and Jing, H. (2003). Discourse segmentation of multi-party conversation. In *Proc. 41st Annual Meeting of the Association for Computational Linguistics (ACL–03)*, pages 562–569.

Garg, A., Pavlovic, V., and Rehg, J. M. (2003). Boosted learning in dynamic Bayesian networks for multimodal speaker detection. *Proc. IEEE*, 91(9):1355–1369.

Geiger, D. and Heckerman, D. (1996). Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence*, 82(1–2):45–74.

Ghahramani, Z. and Jordan, M. I. (1997). Factorial Hidden Markov Models. *Machine Learning*.

Godfrey, J., Holliman, E., and McDaniel, J. (1992). SWITCHBOARD: telephone speech corpus for research and development. In *Proc. IEEE ICASSP*, volume 1, pages 517–520.

Gupta, S., Niekrasz, J., Purver, M., and Jurafsky, D. (2007a). Resolving "you" in multi-party dialog. In *Proc. SIGdial Workshop on Discourse and Dialogue*.

Gupta, S., Purver, M., and Jurafsky, D. (2007b). Disambiguating between generic and referential "you" in dialog. In *Proc. 45th Annual Meeting of the Association for Computational Linguistics (ACL–07)*.

Hain, T., Burget, L., Dines, J., Garau, G., Wan, V., Karafiat, M., Vepa, J., and Lincoln, M. (2007). The AMI system for the transcription of speech in meetings. In *Proc. IEEE ICASSP*, volume 4, pages 357–360.

Hain, T., Garau, G., Karafiát, M., Moore, D., Wan, V., Ordelman, R., and Renals, S. (2005). Transcription of conference room meetings: an investigation. In *Proc. Interspeech - Eurospeech*.

Hakeem, A. and Shah, M. (2004). Ontology and taxonomy collaborated framework for meeting classification. In *Proc. International Conference on Pattern Recognition*, pages 263–268.

Hastie, H., Poesio, M., and Isard, S. (2002). Automatically predicting dialogue structure using prosodic features. *Speech Communication*, 36:63–79.

Howard, A. and Jebara, T. (2004). Dynamical systems trees. In *Proc. International Conference on Uncertainty in Artificial Intelligence*.

Hsueh, P. and Moore, J. (2006). Automatic topic segmentation and labelling in multiparty dialogue. In *Proc. IEEE/ACL Workshop on Spoken Language Technology*.

Hsueh, P. and Moore, J. (2007a). Automatic decision detection in meeting speech. In *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI–07)*, pages 168–179. Springer.

Hsueh, P. and Moore, J. (2007b). What decisions have you made: Automatic decision detection in conversational speech. In *Proc. NACCL–HLT*, pages 25–32, Rochester, NY, USA.

Hsueh, P., Moore, J., and Renals, S. (2006). Automatic segmentation of multiparty dialogue. In *Proc. European Chapter of the Association for Computational Linguistics (EACL)*, pages 273–280.

Huttenhower, C. and Troyanskaya, O. G. (2006). Bayesian data integration: A functional perspective. *Computational Systems Bioinformatics*, 4:341–351.

Janin, A., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Morgan, N., Peskin, B., Pfau, T., Shriberg, E., Stolcke, A., and C.Wooters (2003). The ICSI meeting corpus. In *Proc. IEEE ICASSP*, volume 1, pages 364–367.

Jensen, F. V., Lauritzen, S. L., and Olesen, K. G. (1990). Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly*, 4:269–282.

Ji, G. and Bilmes, J. (2005). Dialog act tagging using graphical models. In *Proc. IEEE ICASSP*, volume 1, pages 33–36.

Jordan, M. I. (1998). *Learning in Graphical Models*. MIT Press.

Jovanovic, N., op den Akker, R., and Nijholt, A. (2006). Addressee identification in face-to-face meetings. In *Proc. Conference of the European Chapter of the Association for Computational Linguistics*, pages 169–176.

Jurafsky, D., Bates, R., Coccaro, N., Martin, R., Meteer, M., Ries, K., Shriberg, E., Stolcke, A., Taylor, P., and Ess-Dykema, C. (1997a). Automatic detection of discourse structure for speech recognition and understanding. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 88–95, Santa Barbara, CA.

Jurafsky, D. and Martin, J. H. (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall.

Jurafsky, D., Shriberg, E., and Biasca, D. (1997b). Switchboard SWBD-DAMSL shallow-discourse-function annotation (coders manual, draft 13). Technical report, Univ. of Colorado, Inst. of Cognitive Science. Available: `http://www.icsi.berkeley.edu/cgi-bin/pubs/publication.pl?ID=001359`.

Jurafsky, D., Shriberg, E., Fox, B., and Curl, T. (1998). Lexical, prosodic, and syntactic cues for dialog acts. In Stede, M., Wanner, L., and Hovy, E., editors, *Discourse Relations and Discourse Markers: Proceedings of the Conference*, pages 114–120. Association for Computational Linguistics, Somerset, New Jersey.

Kadie, C., Hovel, D., and Horvitz, E. (2001). MSBNx: A component-centric toolkit for modeling and inference with Bayesian networks. Technical Report MSR-TR-2001-67, Microsoft Research.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 1(82):35–45.

Kazman, R., Al Halimi, R., Hunt, W., and Mantei, M. (1996). Four paradigms for indexing video conferences. *IEEE Multimedia*, 3(1).

Keizer, S. and op den Akker, R. (2005). Dialogue act recognition under uncertainty using Bayesian networks. *Natural Language Engineering*, 1:1–30.

Khurshid, A. and Denham, S. (2004). A temporal-analysis-based pitch estimation system for noisy speech with a comparative study of performance of recent systems. *IEEE Transactions on Neural Networks*, 15:1112–1124.

Koo, T. and Collins, M. (2005). Hidden-variable models for discriminative reranking. In *Proc. Human Language Technology and Empirical Methods in Natural Language Processing*, pages 507–514.

Kristjansson, T. T., Frey, B. J., and Huang, T. (2000). Event-coupled hidden Markov models. In *Proc. IEEE International Conference on Multimedia and Exposition*, volume 1, pages 385–388.

Küssner, U. (1997). Applying DL in automatic dialogue interpreting. In *Proc. International Workshop on Description Logics*, pages 54–58, Yvette, France.

Kwon, J. and Murphy, K. (2000). Modeling freeway traffic with coupled HMMs. Unpublished notes.

L. Xie, S.-F. Chang, A. D. and Sun, H. (2003). Unsupervised discovery of multilevel statistical video structures using hierarchical Hidden Markov Models. In *Proc. IEEE ICME*.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. International Conference on Machine Learning (ICML)*, pages 282–289.

Lathoud, G. and McCowan, I. (2003). Location based speaker segmentation. In *Proc. IEEE ICASSP*.

Lee, D., Erol, B., and Graham, J. (2002). Portable meeting recorder. *ACM Multimedia*.

Lee, J., Kim, G. C., and Seo, J. (1997). A dialogue analysis model with statistical speech act processing for dialogue machine translation. In *Proc. Spoken Language Translations EACL97 Workshop*, pages 10–15, Budapest, Hungary.

Lesch, S. (2005). Classification of multidimensional dialogue acts using maximum entropy. Diploma thesis, Saarland University.

Lesch, S., Kleinbauer, T., and Alexandersson, J. (2005a). A new metric for the evaluation of dialog act classification. In *Proc. Workshop on the Semantics and Pragmatics of Dialogue (SEMDIAL)*, pages 143–146.

Lesch, S., Kleinbauer, T., and Alexandersson, J. (2005b). Towards a decent recognition rate for the automatic classification of a multidimensional dialogue act tagset. In *Proceedings of the* 4^{th} *IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 46–53, Edinburgh, Scotland, UK.

Levin, L., Langley, C., Lavie, A., Gates, D., Wallace, D., and Peterson, K. (2003). Domain specific speech acts for spoken language translation. In *Proc. SIGdial Workshop on Discourse and Dialogue*, Sapporo, Japan.

Levy, B. C., Benveniste, A., and Nikoukhah, R. (1996). High-level primitives for recursive maximum likelihood estimation. *IEEE Transactions on Automatic Control*, 41(8):1125–1145.

Liu, L., Hu, W., Lai, C., Jiang, H., Chen, W., Zheng, W., and Zhang, Y. (2005). Parallel module network learning on distributed memory multiprocessors. In *Proc. International Conference on Parallel Processing Workshops (ICPPW)*, pages 129–134.

Liu, Y. (2006). Using SVM and error-correcting codes for multiclass dialog act classification in meeting corpus. In *Proc. Interspeech - ICSLP*, pages 1938–1941.

Malkin, J., Xiao, L., and Bilmes, J. (2005). A graphical model for formant tracking. In *Proc. IEEE ICASSP*, volume 1, pages 913–916.

McCowan, I., Bengio, S., Gatica-Perez, D., Lathoud, G., Monay, F., Moore, D., Wellner, P., and Bourlard, H. (2003). Modelling human interaction in meetings. In *Proc. IEEE ICASSP*.

McCowan, I., Gatica-Perez, D., Bengio, S., and Lathoud, G. (2005). Automatic analysis of multimodal group actions in meetings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:305–317.

McGrath, J. E. (1991). Time, interaction and performance (TIP) - a theory of groups. *Small Group Research*, 22(2):116–129.

McNeill, D. and Duncan, S. D. (2000). *Language and Gesture*. Cambridge University Press.

Morgan, N. and Fosler-Lussier, E. (1998). Combining multiple estimators of speaking rate. In *Proc. IEEE ICASSP*, pages 729–732.

Morgan, N., Fosler-Lussier, E., and Mirghafori, N. (1997). Speech recognition using on-line estimation of speaking rate. In *Proc. Eurospeech*, pages 2079–2082.

Mostefa, D., Moreau, N., Choukri, K., Potamianos, G., Chu, S. M., Tyagi, A., Casas, J. R., Turmo, J., Cristoforetti, L., Tobia, F., Pnevmatikakis, A., Mylonakis, V., Talantzis, F., Burger, S., Stiefelhagen, R., Bernardin, K., and Rochet, C. (2007). The CHIL audiovisual corpus for lecture and meeting analysis inside smart rooms. *Language Resources and Evaluation*, 41(3–4):389–407.

Mousset, E., Ainsworth, W. A., and Fonollosa, J. A. R. (1996). A comparision of several recent methods of fundamental frequency and voicing decision estimation. In *Proc. ICSLP*, pages 1273–1276.

Murphy, K. and Weiss, Y. (2001). The factored frontier algorithm for approximate inference in DBNs. In *Proc. UAI*, pages 378–385.

Murphy, K. P. (2002a). *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, U.C. Berkeley, Computer Science Division.

Murphy, K. P. (2002b). Hidden semi-Markov models (HSMMs). Unpublished notes.

Murray, G., Hsueh, P., Tucker, S., Kilgour, J., Carletta, J., Moore, J., and Renals, S. (2007). Automatic segmentation and summarization of meeting speech. In *Proc. NACCL–HLT*, pages 9–10, Rochester, NY, USA.

Murray, G. and Renals, S. (2006). Dialogue act compression via pitch contour preservation. In *Proc. Interspeech*, Pittsburgh, USA.

Murray, G., Renals, S., Carletta, J., and Moore, J. (2006). Incorporating speaker and discourse features into speech summarization. In *Proc. NACCL–HLT*, pages 367–374, New York, USA.

Murray, K. (2001). A study of automatic pitch tracker doubling/halving "errors". In *Proc. SIGdial Workshop on Discourse and Dialogue*, pages 1–4.

Nagata, M. and Morimoto, T. (1993). An experimental statistical dialogue model to predict the speech act type of the next utterance. In *Proc. International Symposium on Spoken Dialogue*, pages 83–86.

Nefian, A. V., Liang, L., Pi, X., Liu, X., Mao, C., and Murphy, K. (2002). A coupled HMM for audio-visual speech recognition. In *Proc. IEEE ICASSP*, pages 2013–2016.

NIST (2004). Spring 2004 (RT-04S) Rich Transcription Meeting Recognition Evaluation Plan. Available: `http://www.nist.gov/speech/tests/rt/2004-spring/documents/rt04s-meeting-eval-plan-v1.pdf`.

Oliver, N. and Horvitz, E. (2003). Selective perception policies for guiding sensing and computation in multimodal systems: a comparative analysis. In *Proc. International Conference on Multimodal Interfaces (ICMI–03)*, pages 36–43.

OpenCV, I. (2001). Open Computer Vision library. Available: `http://sourceforge.net/projects/opencvlibrary/`.

Otsuka, K., Yamato, J., Takemae, Y., and Murase, H. (2006). Conversation scene analysis with dynamic Bayesian network based on visual head tracking. In *Proc. IEEE ICME*, pages 949–952.

Oviatt, S. (2003). Multimodal interfaces. *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, pages 286–304.

Pallett, D. S., Fisher, W. M., and Fiscus, J. G. (1990). Tools for the analysis of benchmark speech recognition tests. In *Proc. IEEE ICASSP*, volume 1, pages 97–100.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 6(2):559–572.

Pevzner, L. and Hearst, M. (2002). A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28:19–36.

Pfau, T., Ellis, D., and Stolcke, A. (2001). Multispeaker speech activity detection for the ICSI meeting recorder. In *Proc. IEEE ASRU Workshop*.

Portinale, L., Bobbio, A., Raiteri, D. C., and Montani, S. (2007). Compiling dynamic fault trees into Dynamic Bayesian Nets for reliability analysis: the `Radyban` tool. In *Proc. Conference on Uncertainty in Artificial Intelligence, Bayesian Modeling Applications Workshop*.

Potamianos, G., Neti, C., Gravier, G., Garg, A., and Senior, A. W. (2003). Recent advances in the automatic recognition of audiovisual speech. *Proc. IEEE*, 91(9):1306–1326.

Potamianos, G., Neti, C., Luettin, J., and Matthews, I. (2004). Audiovisual automatic speech recognition: An overview. *Issues in Visual and Audio-Visual Speech Processing, MIT Press*.

Purver, M., Niekrasz, J., and Ehlen, P. (2007). Automatic annotation of dialogue structure from simple user interaction. In *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI–07)*.

Quattoni, A., Collins, M., and Darrell, T. (2005). Conditional random fields for object recognition. *Advances in Neural Information Processing Systems*, 17:1097–1104.

Rabiner, L. R. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings IEEE*, 2(77):257–286.

Redner, R. A. and Walker, H. F. (1984). Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, 26(2):195–239.

Reiter, S. and Rigoll, G. (2004). Segmentation and classification of meeting events using multiple classifier fusion and dynamic programming. In *Proc. IEEE ICPR*.

Reiter, S. and Rigoll, G. (2005). Multimodal meeting analysis by segmentation and classification of meeting events based on a higher level semantic approach. In *Proc. IEEE ICASSP*, volume 2, pages 161–164.

Reiter, S., Schuller, B., and Rigoll, G. (2007). Hidden conditional random fields for meeting segmentation. In *Proc. IEEE ICME*, pages 639–641.

Renals, S., Hain, T., and Bourlard, H. (2008). Interpretation of multiparty meetings the AMI and AMIDA projects. In *Proc. IEEE Workshop on Hands Free Speech Communication and Microphone Arrays (HSCMA)*, pages 115–118.

Reyes-Gomez, M. J., Raj, B., and Ellis, D. P. W. (2003). Multi-channel source separation by beamforming trained with factorial HMMs. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 13–16.

Rienks, R., Zhang, D., Gatica-Perez, D., and Post, W. (2006). Detection and application of influence rankings in small group meetings. In *Proc. International Conference on Multimodal Interfaces (ICMI–06)*, pages 257–264.

Rosenberg, A. and Hirschberg, J. (2006). Story segmentation of brodcast news in english, mandarin and arabic. In *Proc. HLT–NAACL*, pages 125–128.

Rosset, S. and Lamel, L. (2004). Automatic detection of dialog acts based on multilevel information. In *Proc. ICSLP*, pages 540–543, Jeju Island, Korea. Available: `ftp://tlp.limsi.fr/public/TuB401o.2_p540.pdf`.

Russell, M. J. and Moore, R. K. (1985). Explicit modeling of state occupancy in hidden Markov models for automatic speech recognition. In *Proc. IEEE ICASSP*, pages 5–8.

Schultz, T., Waibel, A., Bett, M., Metze, F., Pan, Y., Ries, K., Schaaf, T., Soltau, H., Westphal, M., Yu, H., and Zechner, K. (2001). The ISL meeting room system. In *Proc. Workshop on Hands-Free Speech Communication*.

Shafer, G. R. and Shenoy, P. P. (1988). Local computation in hypertrees. Technical Report 201, School of Business, University of Kansas.

Shen, L., Sarkar, A., and Och, F. (2004). Discriminative reranking for machine translation. In *Proc. HLT–NAACL*, pages 177–184.

Shi, J. and Tomasi, C. (1994). Good features to track. In *Proc. IEEE CVPR*, pages 593–600.

Shriberg, E., Bates, R., Taylor, P., Stolcke, A., Jurafsky, D., Ries, K., Coccaro, N., Martin, R., Meteer, M., and Ess-Dykema, C. V. (1998). Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech*, 41:439–487.

Shriberg, E., Dhillon, R., Bhagat, S., Ang, J., and Carvey, H. (2004). The ICSI meeting recorder dialog act (MRDA) corpus. In *Proc. HLT–NAACL SIGDIAL Workshop*, pages 97–100.

Shriberg, E., Stolcke, A., Hakkani-Tür, D., and Tür, G. (2000). Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, 32:127–154.

Simhon, S. and Dudek, G. (2004). Sketch interpretation and refinement using statistical models. In *Proc. Eurographics Symposium on Rendering*, pages 23–32.

Skounakis, M., Craven, M., and Ray, S. (2003). Hierarchical Hidden Markov Models for information extraction. In *Proc. International Joint Conference on Artificial Intelligence*.

Smyth, P., Heckerman, D., and Jordan, M. I. (1997). Probabilistic independence networks for hidden Markov probability models. *Neural Computation*, 9(2):227–269.

Snoek, C. G., Worring, M., and Hauptmann, A. G. (2004). Detection of TV news monologues by style analysis. In *Proc. IEEE ICME*.

Sonmez, K., Shriberg, E., Heck, L., and Weintraub, M. (1998). Modelling dynamic prosodic variation for speaker verification. In *Proc. ICSLP*, volume 7, pages 3189–3192.

Stiefelhagen, R. (2002). Tracking focus of attention in meetings. In *Proc. IEEE Conference on Multimodal Interfaces*, pages 273–280.

Stolcke, A. (2002). SRILM an extensible language modeling toolkit. In *Proc. International Conference on Spoken Language Processing*.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Jurafsky, D., Taylor, P., Martin, R., Ess-Dykema, C. V., and Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

Stolcke, A. and Shriberg, E. (1996). Automatic linguistic segmentation of conversational speech. In *Proc. ICSLP*, volume 2, pages 1005–1008.

Stolcke, A., Shriberg, E., Bates, R., Coccaro, N., Jurafsky, D., Martin, R., Meteer, M., Ries, K., Taylor, P., and Ess-Dykema, C. V. (1998). Dialog act modeling for conversational speech. In *Proc. AAAI-98 Spring Symposium on Applying Machine Learning to Discourse Processing*, pages 98–105.

Stolcke, A., Shriberg, E., Hakkani-Tür, D., Tür, G., Rivlin, Z., and Sonmez, K. (1999). Combining words and speech prosody for automatic topic segmentation. In *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, pages 61–64.

Surendran, D. and Levow, G. A. (2006). Dialog act tagging with support vector machines and hidden Markov models. In *Proc. Interspeech - ICSLP*. Article 1831.

Talkin, D. (1995). A robust algorithm for pitch tracking (RAPT). In Kleijn, W. B. and Paliwal, K. K., editors, *Speech Coding and Synthesis*, pages 495–518. Elsevier.

Taylor, P., King, S., Isard, S., and Wright, H. (1998). Intonation and dialog context as constraints for speech recognition. *Language and Speech*, 41:489–508.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.

Venkataraman, A., Ferrer, L., Stolcke, A., and Shriberg, E. (2003). Training a prosody-based dialog act tagger from unlabeled data. In *Proc. IEEE ICASSP*.

Venkataraman, A., Liu, Y., and Shriberg, E. (2005). Does active learning help automatic dialog act tagging in meeting data? In *Proc. Interspeech - Eurospeech*, pages 2777–2780.

Venkataramani, V., Chakrabartty, S., and Byrne, W. (2007). *Gini*support vector machines for segmental minimum Bayes risk decoding of continuous speech. *Computer Speech and Language*, 21(3):423–442.

Verbree, D., Rienks, R., and Heylen, D. (2006). Dialogue-act tagging using smart feature selection; results on multiple corpora. In *IEEE Spoken Language Technology Workshop*, pages 70–73.

Vertegaal, R., Slagter, R., van der Veer, G., and Nijholt, A. (2001). Eye gaze patterns in conversations: there is more to conversational agents than meets the eyes. In *Proc. ACM SIGCHI conference on Human Factors in Computing Systems*, pages 301–308.

Voss, L. and Ehlen, P. (2007). The CALO meeting assistant. In *Proc. HLT–NAACL Demonstration Program*, pages 17–18.

Wahlster, W. (2000). *Mobile Speech-to-Speech Translation of Spontaneous Dialogs: An Overview of the Final Verbmobil System*, pages 3–21. Springer.

Waibel, A., Bett, M., Metze, F., Ries, K., Schaaf, T., Schultz, T., Soltau, H., Yu, H., and Zechner, K. (2001). Advances in automatic meeting record creation and access. In *Proc. IEEE ICASSP*.

Wang, T., Diao, Q., Zhang, Y., Song, G., Lai, C., Bradski, and Bradski, G. (2004). A dynamic Bayesian network approach to multi-cue based visual tracking. In *Proc. ICPR 2004*, volume 2, pages 167–170.

Warner, H. R., Toronto, A. F., Veasey, L. G., and Stephenson, R. (1961). A mathematical approach to medical diagnosis – application to congenital heart disease. *Journal of the American Medical Association*, pages 177–183.

Warnke, V., Kompe, R., Niemann, H., and Nöth, E. (1997). Integrated dialog act segmentation and classification using prosodic features and language models. In *Proc. Interspeech - Eurospeech*, volume 1, pages 207–210.

Weiland, M., Smaill, A., and Nelson, P. (2005). Learning musical pitch structures with Hierarchical Hidden Markov Models. In *Proc. Journes d'Informatique Musicale*, pages 55–61.

Wellman, M. P. and Henrion, M. (1993). Explaining "Explaining Away". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):287–292.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques (2nd Edition)*. Morgan Kaufmann, San Francisco.

Wrigley, S., Brown, G., Wan, V., and Renals, S. (2005). Speech and crosstalk detection in multi-channel audio. *IEEE Transactions on Speech and Audio Processing*, 13(1):84–91.

Xie, L. and Liu, Z.-Q. (2007). A coupled HMM approach to video-realistic speech animation. *Pattern Recognition*, 40(8):2325–2340.

Yang, J., Lu, W., and Waibel, A. (1998). Skin-color modeling and adaptation. In *Proc. of Asian Conference on Computer Vision*, volume 2, pages 687–694.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. (2006). *The HTK book*. Cambridge University Engineering Department.

Zhang, D., Gatica-Perez, D., Bengio, S., and McCowan, I. (2006). Modeling individual and group actions in meetings with layered HMMs. *IEEE Transactions on Multimedia*, 8:509–520.

Zhang, D., Gatica-Perez, D., Bengio, S., McCowan, I., and Lathoud, G. (2004a). Modeling individual and group actions in meetings: a two-layer HMM framework. In *Proc. IEEE CVPR, Workshop on Event Mining in Video*.

Zhang, D., Gatica-Perez, D., Bengio, S., McCowan, I., and Lathoud, G. (2004b). Multimodal group action clustering in meetings. In *Proc. ACM Multimedia, Workshop on Video Surveillance and Sensor Networks*.

Zhang, N. L. and Poole, D. (1994). A simple approach to Bayesian network computations. In *Proc. Canadian Conference on Artificial Intelligence*, pages 171–178.

Zhang, Y., Diao, Q., Huang, S., and Hu, W. (2003). DBN based multi-stream models for speech. In *Proc. IEEE ICASSP*.

Zhong, S. and Ghosh, J. (2002). HMMs and coupled HMMs for multi-channel EEG classification. In *Proc. IEEE Joint Conference on Neural Networks*, volume 2, pages 1154–1159.

Zimmermann, M., Liu, Y., Shriberg, E., and Stolcke, A. (2005). A* based joint segmentation and classification of dialog acts in multiparty meetings. In *Proc. IEEE ASRU*, pages 215–219.

Zimmermann, M., Liu, Y., Shriberg, E., and Stolcke, A. (2006a). Toward joint segmentation and classification of dialog acts in multiparty meetings. In *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI–05)*, pages 187–193. Springer.

Zimmermann, M., Stolcke, A., and Shriberg, E. (2006b). Joint segmentation and classification of dialog acts in multiparty meetings. In *Proc. IEEE ICASSP*, volume 1.

Zweig, G. (1996). A forward-backward algorithm for inference in Bayesian networks and an empirical comparison with HMMs. Masters thesis, Dept. Computer Sciences, U.C. Berkeley.

Zweig, G. (1998). *Speech Recognition with Dynamic Bayesian Networks*. PhD thesis, Dept. Computer Sciences, U.C. Berkeley.

Zweig, G. and Russel, S. J. (1998). Speech recognition with Dynamic Bayesian Networks. In *Proc. Artificial Intelligence/Innovative Applications of Artificial Intelligence*, pages 173–180.